(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 May 2001 (03.05.2001)

PCT

(10) International Publication Number
WO 01/31900 A2

(51) International Patent Classification[7]: H04M 3/00

(21) International Application Number: PCT/US00/29576

(22) International Filing Date: 26 October 2000 (26.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/161,444 26 October 1999 (26.10.1999) US
60/190,613 20 March 2000 (20.03.2000) US

(71) Applicant: PINGTEL CORPORATION [US/US]; Suite 2200, 400 West Cummings Park, Woburn, MA 01801 (US).

(72) Inventors: BATSON, James, A., Jr.; 773 Winter Street, North Andover, MA 01845 (US). PETRIE, Daniel, G.; 34 Robbins Road, Arlington, MA 02476 (US). SCHAAF, Richard, W.; 583 North Road, Sudbury, MA 01776 (US).

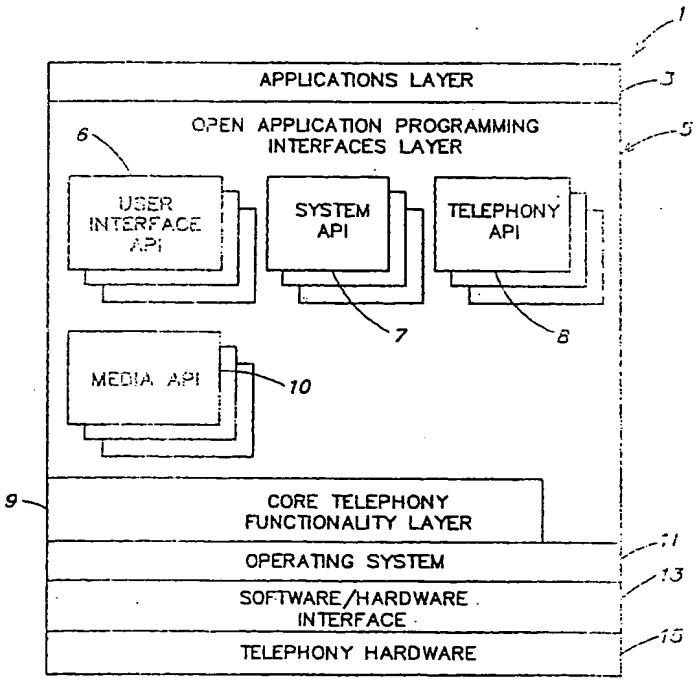
(74) Agent: MCLOUGHLIN, Daniel, P.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: DISTRIBUTED COMMUNICATIONS NETWORK INCLUDING ONE OR MORE TELEPHONY COMMUNICATION DEVICES HAVING PROGRAMMABLE FUNCTIONALITY



(57) Abstract: A computer program product having an open telephony system architecture such that one or more telephony applications defined on the computer program product may be modified, after initial deployment in the field (e.g. on a customer premise), independently of a vendor that controlled development of the one or more telephony applications. Further, a computer program product having an extensible telephony system architecture such that the telephony functionality defined thereon can be expanded by adding telephony applications to the

[Continued on next page]

BEST AVAILABLE COPY

WO 01/31900 A2

IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

computer program product. Also, a computer program product capable of controlling a connection during a telephone call using any of a plurality of call control protocols, including SIP, H.323, MGCP, Megaco/H.248, and the Skinny Station protocol. For a telephone call involving multiple connections, for example, a conference call, the computer program product can control communications on each connection concurrently and can use a different call control protocol for each connection. A communications network including one or more computer program products having one or more of the following properties: an open telephony system architecture, an extensible telephony system architecture; the capability to control a telephony call using any of a plurality of call control protocols.

DISTRIBUTED COMMUNICATIONS NETWORK INCLUDING ONE OR MORE TELEPHONY COMMUNICATION DEVICES HAVING PROGRAMMABLE FUNCTIONALITY

RELATED APPLICATION

This application claims priority under 35 U.S.C. §119(e) to commonly-owned, co-pending U.S. provisional patent applications serial no. 60/161,144, entitled, "Communications Network Using Intelligent Telephones", filed October 26, 1999, and serial no. 60/190,613, entitled, "Method and Apparatus for Organizing Configuration Information for a Communications System" filed March 20, 2000, which both are incorporated herein by reference in their entities.

FIELD

This application is directed to a communications network having intelligent end points for placing telephone calls. In particular, this application relates to a data network including one or more telephony communication devices that provide and control one or more telephony applications.

BACKGROUND

For a typical telephony communications network, including wireless and wire-based networks, the endpoints of such a network are telephony communication devices such as a telephone or a telephony-enabled computer.

A telephony communication device as used herein is a device capable of performing at least the following traditional telephony tasks: initiating the setting up of a telephone call by sending a signal onto a transmission medium of a communications network, detecting and indicating (e.g., ringing, beeping or blinking a light) that a telephone call is incoming, determining that a user has responded to a phone call (e.g., off-hook detection, a keypad entry, a keyboard entry or a mouse click), sending a signal indicating that the user has responded onto the transmission medium, receiving dialing instructions from a user (e.g., from a rotary dial, push buttons, voice activation, keyboard or mouse), sending a signal representing the dialing instructions on to the transmission medium, transmitting and receiving acoustic audio signals to

and from, respectively, a user, and transmitting and receiving media (e.g., audio data) on to the transmission medium of the network.

As used herein, a "telephony function" is a function performed in association with a telephone call. A "telephony application" is an application that performs one or more telephony functions. Telephony functions may be divided into two categories: core telephony functions and feature telephony functions. Core telephony functions may be divided into three categories: phone set management, call control and media processing.

As used herein, "phone set management" refers to the control of low-level device interactions on a telephony communication device such as, for example, button presses, hook switch operations and lamp or light emitting diode (LED) operations.

As used herein, "media processing" refers to the transmitting and receiving of media between a user of a telephony communication device and a transmission medium of a network on which the telephony communication device resides.

As used herein, "call control" refers to the controlling of setting up, maintaining and tearing down a telephone call. For a traditional telephony communication device, controlling a telephone call typically involves the use of a single call control protocol including peer-to-peer-type protocols such as, for example, the Sessions Initiation Protocol (SIP) or the H.323 protocol, or a master/slave-type protocol such as, for example, the Media Gateway Control Protocol (MGCP), the Megaco/H.248 protocol, or the Skinny Station protocol promulgated by Cisco Systems, Inc.

The SIP protocol is defined in RFC 2543, SIP: Session Initiation Protocol by Internet Engineering Task Force (IETF) as of October 26, 1999. The H.323 protocol is described by the International Telecommunications Union in the ITU-TN Recommendation: H.323 Packet-Based Multi-Media Communications System, Geneva, Switzerland, February, 1998. The Megaco/H.248 protocol is defined by IETF RFC 2885 and ITU H.248. The MGCP protocol is described in RFC 2705 as of October 1999.

As used herein, a "feature telephony function" is a function performed in association with a telephone call, beyond or in enhancement of the core telephony functions provided by phone set management, media processing and call control. For example, traditional telephony feature functions include placing a call on hold, call forwarding, voice mail functions, call waiting, conference calling, etc... As referred to

herein, a "feature telephony application" is an application that performs one or more feature telephony functions.

Performing core telephony functions and feature telephony applications requires processing resources. In traditional telephony communications networks such as, for example, Public Switched Telephone Networks (PSTN), Private Branch Exchanges (PBX), or Central Office Exchange Services (Centrex), telephony communication devices rely on a server, switch, or other centralized processing entities as the processing resource to control and perform the bulk of the telephony functions associated with a telephone call. Thus, traditional telephony communications networks typically have centralized processing resources.

In typical traditional communications networks, to perform telephony functions, a telephony communication device receives instructions from one or more network resources such as a server or central switch, and operates based on those instructions. Typically, the network resource maintains state information associated with the telephony functions and decides how to respond to events associated with the functions.

Traditional communications networks include analog lines for transmitting analog data, digital lines for transmitting digital data, or a combination of both analog and digital lines. Further, some of these networks may include high-speed backbone segments, for example, a backbone segment comprising a fiber optic cable.

Further, traditional communications networks may include any of a variety of telephones including telephones that include analog logic for performing telephony functions, digital logic for performing telephony functions or a combination of analog and digital logic.

For a telephone connected to a digital line, the telephone must include logic for converting acoustic audio signals into digital audio data. As used herein, a "digital telephony communication device" is a telephony communication device that, in addition to the telephony tasks described above, converts acoustic audio signals into digital audio data, and transmits digital data, including digital audio data, onto a data line.

More recently, communications networks have begun using data networks to transmit voice data (e.g., Voice-over-IP data) and other data between a caller and one or more recipients. Such data networks include Local Area Networks (LANs), Metropolitan Area Networks (MANs) and Wide Area Networks (WANs) such as the

Internet. Typically, these data networks are designed to transmit data in accordance with the Internet Protocol (IP).

During a telephone call across one or more data networks, digital audio data from a digital telephony communication device on a data network may be transmitted to another digital communication telephony device network on the data network, or to gateways between the data network and other communications networks.

Data networks typically are more distributed (i.e., less centralized) than traditional telephony communications networks, relying on one or more servers or other network resources for processing as opposed to a central switch.

In data networks, one or more telephony functions and/or applications may be made available on a digital telephony communication device through hardware, sometimes in combination with software and/or firmware. Typically, after initial deployment (i.e., installation) of the communication device in the field, e.g., at a customer's premises, the available telephony functions are essentially fixed in nature. Sometimes, one or more of the telephony functions include one or more user-definable parameters that allow a user to define values, and thus have a limited role in programming the telephony communication device. For example, user-definable parameters may include the type or volume of a ring, a user password, numbers to be dialed automatically at the press of a single button, etc. The telephony functions themselves, however, are "closed" (i.e., inaccessible) to a user of the telephony communication device and third party vendors such that the definition of the telephony functions may not be modified by the user or third party vendors.

Typically, to change a telephony function on an installed digital telephony communication device, the vendor that controlled development of the telephony functions must provide new hardware or software, or upgrade the existing hardware or software.. Further, additional telephony applications may not be added to a telephony communication device after the telephony communication device has been installed in the field.

The telephony capabilities of typical telephony communication devices are limited in several ways. First, many telephony communication devices today are not capable of controlling telephone calls in which they participate, and those devices that are capable of controlling a telephone call (e.g., some digital phones) are capable of

controlling the telephone call using only a single call control protocol, for example, SIP or H.323.

Second, many telephony communication devices perform telephony operations in response to instructions received from external communications network resources upon which telephony applications are being executed. Further, for those telephony communication devices upon which telephony applications are defined and upon which the telephony applications may be executed, after these telephony communication devices have been initially deployed in the field, no further telephony applications may be added to the telephony communication device.

Also, for such telephony communication devices, after these telephony communication devices have been deployed in the field, the telephony applications are closed to users and third party vendors such that the telephony applications are essentially fixed in nature, i.e., they may not be modified by users and third party vendors.

The telephony capabilities of typical communications networks also are limited in several ways, due in part to the limitations of the telephony communication devices that reside on these communication networks.

First, the processing resources of many communications networks are highly centralized such that relatively few network resources provide telephony functionality for the entire communications network. Second, even for those communication networks that are more distributed and include one or more telephony communication devices that have telephony applications defined thereon and have the capability to execute such telephony applications, after a telephony communication device has been initially deployed, its telephony applications are not modifiable by a user or a third party vendor and additional telephony applications may not be added to the telephony communication device.

Consequently, the telephony functionality of a traditional communications network is limited either by the telephony functionality provided by a centralized network resource or the fixed telephony functionality available on the telephony communication devices of the network. These limitations impair the flexibility of a network user to expand or modify the telephony functionality of the user's telephony communication device, thereby limiting the scalability and flexibility of the communications network.

SUMMARY

Accordingly, provided herein is a telephony communication device having an open telephony system architecture such that one or more telephony applications defined on the telephony communication device may be modified, after initial deployment in the field (e.g., on a customer premise), independently of a vendor that controlled development of the one or more telephony applications.

Also, provided herein is a telephony communication device having an extensible telephony system architecture such that the telephony functionality defined thereon can be expanded by adding telephony applications to the telephony communication device.

Also provided herein is a telephony communication device capable of controlling a connection during a telephone call using any of a plurality of call control protocols, including SIP, H.323, MGCP, Megaco/H.248, and the Skinny Station protocol. Further, for a telephone call involving multiple connections, for example, a conference call, the telephony communication device can control communications on each connection concurrently and can use a different call control protocol for each connection.

Also provided herein is a communications network including one or more telephony communication devices having one or more of the following properties: an open telephony system architecture, an extensible telephony system architecture; the capability to control a telephony call using any of a plurality of call control protocols. Such a communications network has more flexible and scaleable telephony functionality than existing communication networks.

In a first embodiment, provided is a first telephony communication device that is part of a communications network that includes a transmission medium. The first telephony communication device includes a telephony hardware component and a telephony software component. The telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more inputs to transmit media to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. The telephony application is modifiable after the first telephony communication device is deployed on

the communications network with modifications developed independently from creation of the telephony software component.

In another embodiment, a method of defining functionality for a telephony communication device is provided, where the telephony communication device is part of a communications network that includes a transmission medium. The telephony communication device includes a telephony hardware component and a telephony software component. The telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and includes one or more outputs to transmit media to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. After the telephony communication device is deployed on the communications network, the telephony application is accessed on the telephony communication device, and modified with modifications developed independently from creation of the telephony software component.

This embodiment may be implemented as a computer program product that includes a computer-readable medium and computer-readable signals stored on the computer-readable medium that define instructions. These instructions, as a result of being executed by a computer, instruct the computer to perform the acts described above for this embodiment.

In yet another embodiment, provided is a system for defining functionality for telephony communication device that is part of a communications network that includes a transmission medium. The telephony communication device includes a telephony hardware component and a telephony software component. The telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and includes one or more outputs to transmit media to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. The system includes means for accessing the telephony application on the telephony communication device after the telephony communication device is deployed on the communications network, and means for modifying the

telephony application with modifications developed independently from creation of the telephony software component.

In another embodiment, a communication network is provided that includes a transmission medium and one or more telephony communication devices, where each telephony communication device comprises a telephony hardware component and a telephony software component. For each telephony communication device, the telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. The telephony application is modifiable after the telephony communication device is deployed on the communication network with modifications developed independently from creation of the telephony software component.

In another embodiment, provided is a first telephony communication device that is part of a communications network that includes a transmission medium, where the first telephony communication device includes a telephony hardware component and a telephony software component. The telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and includes one or more outputs to transmit media to the first user and data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. At least part of an additional telephony application can be added to the telephony software component after the first telephony communication device is deployed on the communications network.

In yet another embodiment, functionality is defined for a telephony communication device that is part of a communication network that includes a transmission medium, where the telephony communication device includes a telephony hardware component and a telephony software component. The telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and includes one or more outputs to transmit media

to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. After the telephony communication device is deployed on the communication network, the telephony software component may be accessed on the telephony communication device, and at least a part of an additional telephony application may be added to the telephony software component.

This embodiment may be implemented as a computer program product that includes a computer-readable medium and computer-readable signals stored on the computer-readable medium that define instructions. These instructions, as a result of being executed by a computer, instruct the computer to perform the acts described above for this embodiment.

In another embodiment, provided is a system for defining functionality for a telephony communication device that is part of a communications network that includes a transmission medium, where the telephony communication device includes a telephony hardware component and a telephony software component. The telephony hardware component includes one or more inputs to receive audio input from a first user and first data from the transmission medium, and includes one or more outputs to transmit media to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. This system includes means for accessing the telephony software component after the telephony communication device is deployed on the communications network, and means for adding at least a part of an additional telephony application to the telephony software component.

In another embodiment, a communications network is provided that includes a transmission medium and one or more telephony communication devices, where each telephony communication device includes a telephony hardware component and a telephony software component. For each telephony communication device, the telephony hardware component comprises one or more inputs to receive audio input from a first user and first data from the transmission medium, and includes one or more outputs to transmit media to the first user and second data to the transmission medium. The telephony software component controls operation of the hardware component and

includes at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call. At least a part of an additional telephony application can be added to the telephony software component after the telephony communication device is deployed on the communications network.

In yet another embodiment, provided is a first telephony communication device that is part of a communication network that includes a transmission medium, where the telephony communication device includes a call processing module. The call processing module represents and controls at least a first telephone call that includes one or more connections to other users, where each user corresponds to another telephony communication device on the communications network. For at least a first of the one or more connections, the call processing modules operative to control communications on a telephony device corresponding to the first connection using a first call control protocol selectable from a plurality of call control protocols available on the first telephony communication device.

In another embodiment, a first telephone call is controlled on a first telephony communication device that is connected to a communications network. The telephone call includes one or more connections to other users, where each user corresponds to another telephony communication device on the communications network. For at least a first of the one or more communications, a first call control protocol is selected from a plurality of call control protocols available on the first telephony communication device, and communications on the telephony communication device corresponding to the first connection are controlled using the first call control protocol.

This embodiment may be implemented as a computer program product that includes a computer-readable medium and computer-readable signals stored on the computer-readable medium that define instructions. These instructions, as a result of being executed by a computer, instruct the computer to perform the acts described above for this embodiment.

In another embodiment, provided is a system for controlling a first telephone call on a first telephony communication device connected to a communications network. The telephone call includes one or more connections to other users, where each user corresponds to another telephony communication device on the communications network. For at least a first of the one or more connections, the system includes means for selecting a first call control protocol from a plurality of call control protocols

available on the first telephony communication device, and means for controlling communications corresponding to the first connection on the telephony communication device using the first call control protocol.

In yet another embodiment, a communications network is provided that includes a transmission medium and one or more telephony communication devices, where each telephony communication device includes a call processing module. For each telephony communication device, the call processing module represents and controls at least a first telephone call that includes one or more connections to other users, where each user corresponds to another telephony communication device on the communications network. For at least a first of the one or more connections, the call processing module is operative to control communications corresponding to the first connection on the telephony communication device using a first call control protocol selectable from a plurality of call control protocols available on the first telephony communication device.

In another embodiment, a computer program product is provided that includes a computer-readable medium, and computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a telephony communication device, instruct the telephony communication device to perform a telephony application. At least a portion of the telephony application may be added to the telephony communication device after the telephony communication device has been deployed on a communications network.

In another embodiment, a computer program product is provided that includes a computer-readable medium, and computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a telephony communication device, instruct the telephony communication device to perform a telephony application. At least a portion of the telephony application may be modified on the telephony communication device after the telephony communication device has been deployed on a communications network with modifications developed independently from creation of the telephony application.

The features and advantages of the invention described above and other features and advantages of the invention will be more readily understood and appreciated from the detailed description below, which should be read together with the accompanying drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings,

Fig 1. is a block diagram illustrating an example embodiment of an open and extensible telephony system architecture for a telephony communication device;

Fig. 2 is a block diagram illustrating an example embodiment of a core telephony functionality layer of the open and extensible telephony system architecture of Fig. 1;

Fig. 3 is a data flow diagram illustrating an example embodiment of a media processing component corresponding to the media processing module of Fig. 2;

Fig. 4 is a block diagram illustrating example embodiments of the call processing module of Fig. 2 and the media processing module of Fig. 2;

Fig. 5 is a diagram illustrating an example embodiment of a telephony communication device;

Fig. 6 is a diagram illustrating an example embodiment of a graphical display provided by a telephony communication device;

Fig. 7 is a block diagram illustrating an example embodiment of a communications network including one or more telephony communication devices having modifiable and expandable functionality;

Fig. 8 is a flowchart illustrating an example embodiment of a method of indicating an incoming call to a second user of a telephony communication device;

Fig. 9 is a flowchart illustrating an example embodiment of a method of selectively transmitting media during a telephone call;

Fig. 10 is a flowchart illustrating another example embodiment of a method of selectively transmitting media during a telephone call;

Fig. 11 is a flowchart illustrating an example embodiment of a method of scheduling and performing a telephone call;

Fig. 12 is a flow chart illustrating an example method of displaying text associated with a telephone call;

Fig. 13 is a flow chart illustrating an example embodiment of a method of communicating information in accordance with a state of a telephony communication device;

Fig. 14 is a flow chart illustrating an example embodiment of a method of communicating information in accordance with a state of a telephony communication device;

Fig. 15 is a flow chart illustrating an example embodiment of a method of communicating an audible expression during a telephone call;

Fig. 16 is a flow chart illustrating an example embodiment of a method of screening a telephone call based on information associated with the call;

Fig. 17 is a flow chart illustrating an example embodiment of a method of playing content of an e-mail message as audio on a telephony communication device;

Fig. 18 is a flow chart illustrating an example embodiment of a method of sending a graphical message for a user to a telephony communication device;

Fig. 19 is a flow chart illustrating an example embodiment of a method of communicating a graphical message to a user of a telephony communication device;

Fig. 20 is a flow chart illustrating an example embodiment of a method of placing one or more connections on hold during a telephone call; and

Fig. 21 is a flow chart illustrating an example embodiment of a method of dynamically configuring a telephony communication device.

DETAILED DESCRIPTION

1. Telephony System Architecture

A telephony system architecture is a system architecture including one or more inter-related system components that together define the available telephony applications on a Telephony Communication Device (TCD). An open telephony system architecture is a telephony system architecture that, in addition to a vendor that controlled development of the one or more telephony applications defined by the telephony system architecture, is accessible for users and third party vendors to modify the telephony functionality of one or more of the telephony applications after such telephony applications have been deployed in the field on a TCD. An extensible telephony system architecture is a system architecture that allows telephony applications to be added to a TCD already deployed in the field.

Fig. 1 is a block diagram illustrating an example embodiment of an open and extensible telephony system architecture 1 for a TCD. The extensible telephony system architecture may be considered a layered architecture 1 where adjacent layers of abstraction (or layers) communicate between each other such that layers on either side of a given layer are independent of each other. In other words, functions may be defined for a higher layer of abstraction to be performed on a non-adjacent lower level of abstraction without knowing or specifying the details of the non-adjacent lower layer.

For illustrative purposes, the TCD on which the telephony system architecture 1 resides is referred to herein as a first TCD. This first TCD may be connected to a communications network, as will be described in more detail below in relation to Fig. 7.

The open and extensible telephony system architecture 1 may include an applications layer 3, an open application programming interface layer 5, a core telephony functionality layer 9, an operating system layer 11, a software/hardware interface layer 13 and a telephony hardware layer 15.

1.1 Telephony Hardware Layer

The telephony hardware layer or telephony hardware component 15 includes several hardware elements for performing the telephony operations associated with a telephone call. The telephony hardware component 15 may include a processor for processing instructions corresponding to a telephony operation. Such a processor may be chosen such that the processor is capable of processing a telephone call at a real-time rate such that a user participating in a telephone call does not experience any delays in the

telephony functions associated with the telephone call, particularly in the transmission of audio data. For example, the processor may be the StrongArm 1110 running at 206 MegaHertz (MHz).

The telephony hardware component 15 also may include memory for storing information. For example, the memory may include non-volatile memory such as a flash read-only memory (Flash ROM) for persistently storing telephony applications and data, and may also include volatile memory such as synchronous dynamic random-access memory (SDRAM) for temporary storage of and faster access to telephony applications and data.

The capacities of the volatile memory and non-volatile memory may be selected and changed in accordance with the storage needs and configuration of the first TCD within the communications network. For example, the first TCD may be configured, as a part of being initialized, to download telephony applications and/or non-telephony applications from one or more storage resources on the network. In such a configuration, the first TCD may have a relatively larger amount of volatile memory for faster processing because a relatively lower amount of non-volatile memory is needed for storing applications.

In contrast, the first TCD may be configured to store large amounts of data such as, for example, audio files, or to store one or more large applications or a large number of applications. In such a configuration, the first TCD may have a relatively larger amount of non-volatile memory to store the data and applications, resulting in less volatile memory. Obviously, the capacities of the volatile and non-volatile memories also are affected by the physical space available within the first TCD.

The telephony hardware component 15 also may include audio processing circuitry for processing the audio information transmitted during a telephone call. The audio processing circuitry may include a codec for performing typical audio processing functions including the compression and decompression of audio data being transmitted to and received from, respectively, the communications network. The audio processing module may be designed to sample audio data in a variety of sizes and process the audio data at any of a variety of sample rates. For example, the audio processing module may include the UDA 1341TS stereo codec from Philips Semiconductors and sample audio data at a 16-bit size and at a sample rate of 32 kilohertz (KHz). Other sample rates may be used, such as, for example 44.1 and 48 KHz.

The telephony component 15 also may include video processing circuitry, including one or more video codecs, for processing video using known techniques.

As will be discussed in more detail below in relation to Fig. 5, the telephony hardware component 15 also may include a display screen and associated graphical logic. In an embodiment, the display screen is a 160 x 160 pixels graphics display. The display screen may be capable of displaying color, monochrome, or gray-scale pixels. For example, in an embodiment, the display screen is designed to display pixels using a 15-color gray-scale.

The telephony hardware component 15 also may include a network interface for communicating data between the first TCD and the transmission medium of the communications network. In an embodiment, the communications network adheres to the Ethernet protocol and the network interface is an Ethernet interface such as, for example, the CS 8900A-CQ3 10BaseT Ethernet controller from Cirrus Logic.

The telephony hardware component 15 also includes power circuitry for powering the first TCD. In an embodiment, the first telephony hardware component 15 includes circuitry to receive power supplied by a Category 5 (Cat5) cable. The power supply circuitry may use powering approaches compatible with Cisco Inline-Power techniques for Cat5/Ethernet LANs developed by Cisco Systems, Inc.

1.2 Software/Hardware Interface Layer

The software/hardware interface layer 13 may define a plurality of drivers for operating the hardware elements of the telephony hardware layer 15. The drivers may be programmed in any of a plurality of programming languages, for example, C.

1.3 Operating System Layer

The operating system layer 11 may comprise any of the plurality of operating systems. As will be described below in more detail in relation to Fig. 2, the core telephony functionality layer is configured such that abstraction layers 1, 3, and 5 may be programmed independently of the operating system of the operating system layer 11.

The operating system of the operation system layer 11 may be a real-time operating system (RTOS) or another operating system, such as, for example, Windows® NT, Mac® OS, UNIX® or LINUX®, which are more commonly associated with a general purpose computer.

In an embodiment of the operating system layer 11 that uses an RTOS, the RTOS may be VxWorks® from Wind River Systems, Inc. and the operating system layer 11 may also include other components supplied by Wind River including: Personal JWorks® 3.0 (Personal Java, JDK 1.1.6); True FFS Flash file system manager; the Simple Network Management Protocol (SNMP) v1/v2c; and the Rogue Wave tools.h++ Library.

1.4 Core Telephony Functionality Layer

The core telephony functionality layer 9 defines the core telephony functions associated with a telephone call. Fig. 2 is a block diagram illustrating an example embodiment of the core telephony functionality layer 9, which includes an OS abstraction layer 33, the core telephony functions module 25, the telephony application objects module 23 and the core telephony functionality API module 21.

1.4.1 OS Abstraction Layer

The OS abstraction layer 33 interfaces the native operating system of the operating system layer 11 with the remaining modules and layers of the telephony system architecture, including modules 21, 23 and 25 and layers 3, and 5. The OS abstraction layer 33, by providing an interface to the native operating system (e.g., VxWorks or Windows NT) avoids dependence on the native operating system by the remaining modules and layers. The OS abstraction layer 33 may include abstractions for maintaining date and time information, registering events, managing message queues, managing socket-based communications, synchronizing telephony operations on the TCD, managing tasks, and providing timers.

1.4.2 Core Telephony Functionality Module

The core telephony functions module 25 includes the phone set management module 27, the media processing module 29 and the call processing module 31. The phone set management module 27 includes abstractions defined to handle low-level device interactions, such as button presses, hook switch operations, and lamp (LED) control.

1.4.2.1 Media Processing Module

The media processing module 29 provides a framework for the first TCD to perform real-time audio processing for one or more telephone calls. The media processing module 29 may be organized such that the media data flows through with

minimum latency. The audio data may be processed in chunks, each chunk being audio data from a particular temporal interval. For example, the media processing module 29 may process the audio data in 10 millisecond (ms) chunks. For example, as is described below in more detail in relation to Fig. 3, each processing element of a media processing component 30 may process media in 10 ms chunks.

The media processing module 29 may include one or more media processing components, each media processing component corresponding to a particular telephone call. In a hardware embodiment, the media processing module 29 may include physically-separate media processing components such that the number of telephone calls that the media processing module can process concurrently is limited by the number of these physical media processing components. In such a hardware embodiment, at least part of the media processing module 29 may be implemented using a dedicated digital signal processor (DSP).

In a software embodiment where an object-oriented programming language, for example, C++ or Java, is used, one or more abstractions (e.g., classes or structs) may be used to represent the media processing module 29 and one or more media processing components, as described below in more detail in relation to Fig. 4.

1.4.2.1.1 Media Processing Component

Fig. 3 is a data flow diagram illustrating an example embodiment of a media processing component 30. Traditionally, only audio has been associated with a telephone call. More recently, however, video has become a part of telephone calls. As described below in more detail, the first TCD may support telephone calls including audio, video and other associated data. Accordingly, although most of the media data and media processing elements described below are audio-related, the media processing component 30 also may include video processing elements that process video data in coordination with the audio processing elements of the processing component 30.

The media processing component 30 may include one or more connection media input interfaces 41, a user media input interface 59, a local call bridge 51, one or more connection media output interfaces 87 and a user media output interface 82.

For a telephone call between one or more first users of the first TCD and one or more second users at one or more second TCDs, for each second TCD, the call processing module 31 (described in more detail below in relation to Fig. 4) maintains a corresponding connection to the second TCD. The media processing of the telephone

call is represented by the media processing component 30. Each connection media interface 41 and corresponding connection media output interface 87 together represent a connection between the first TCD and one of the second TCDs.

Each network connection media input interface 41 includes an input buffer 42, a dejitter buffer 44 and a decoder 47.

The input buffer 42 receives connection input media 40 and generates connection media 43. The connection input media 40 is media data received from the transmission medium of the network over a network connection from another telephone call participant, and may be received in the form of data packets conforming to one or more media transport protocols, such as, for example, the Real-Time Transport Protocol (RTP) and the Real-Time Transport Control Protocol (RTCP). The input buffer 42 may de-package the connection input media 40 by removing the media transport information from the audio data packets of the connection input media 40 to produce the connection media 43.

The connection input media 40 may be a chunk data from a temporal interval greater than the temporal interval, e.g. 10 ms, that the media processing component 30 is configured to process. Accordingly, the input buffer 42 may be configured to buffer the connection input media 40, process the input connection media 40 in chunks corresponding to the configured temporal interval, and output the connection media 43 in chunks corresponding to the configured temporal interval. Although not shown in Fig. 3, the other inputs, microphone audio data 55, tone indicator 57 and stored audio data 58, also may be received by input buffers and buffered such that the inputs received by the echo suppressor/cancellor 61, the tone generator 69 and the first mixer 74 are each a chunk of data corresponding to the configured temporal interval.

The dejitter buffer 44 receives the connection media 43 and generates dejittered media 43. The dejitter buffer 44 applies known techniques to remove information from the connection media 43 to remove "jitter" from the resulting media played to a user of the first TCD. During transport across the network medium of the communications network, data packets of the connection input media 40 may have been routed and copied such that the connection input media 40 may include re-ordered packets and/or multiple same packets. For example, if ordered packets 1, 2 and 3 were sent from a second TCD to the first TCD, packet 1 may be duplicated and the packets re-ordered such that packets 1, 3, 1 and 2 arrive at the first TCD. The dejitter buffer 44 of the TCD includes logic to

recognize and eliminate re-ordered and duplicate packets such that the dejittered media 45 includes the originally transmitted media packets 1, 2 and 3, properly ordered and without duplicates.

The connection input media may have been encoded using known techniques by a second TCD before transmission to the first TCD. The decoder 47 receives the dejittered media 45, and decodes the dejittered media 45 using known techniques to produce decoded connection media 49, which is then sent to local call bridge 51.

The user media input interface 59 includes an echo suppressor/canceller 61, a tone generator 69, a first mixer 74, a second mixer 65 and a splitter 73.

The echo suppressor/canceller 61 receives microphone audio data 55 and generates de-echoed audio data 63. The microphone audio data 55 is audio data received from the user of the TCD through a microphone. For example, the microphone audio data 55 may be received from a microphone of a telephone handset, a microphone embedded in a telephone base as part of a speaker phone, or a microphone used for input to a computer.

The echo suppressor/canceller may include echo suppression logic to control the suppression of echoes in the microphone audio data 55. For example, if a TCD has a speaker phone base assembly including a base speaker for playing audio signals of a telephone call, the echo suppressor/canceller 61 may include echo suppression logic to detect when the microphone audio data 55 includes voice data and to turn off the base speaker as a result of this detection. Accordingly, the microphone that a user is using during a telephone call does not receive any sound from the base speaker which would cause echoes of one or more voices of the telephone call to be included in the microphone audio data 55.

The echo suppressor/canceller 61 also may include echo cancellation logic to apply echo cancellation techniques, by recognizing the lower intensity, delayed audio signals of an echo and to subtract these signals from the microphone audio data 55 to produce the de-echoed audio data 63. Other echo suppression and cancellation techniques may be used.

The tone generator 69 receives a tone indicator 57 and generates an indicated tone 71. The indicated tone 71 is a tone indicated by the tone indicator 57. For example, the tone indicator 57 may represent a digit of a telephone number entered by a user from a standard telephony touch tone dialing keypad, computer keyboard or mouse. In this

example, the indicated tone 71 generated by the tone generator 69 may be a tone associated with the digit in accordance with the Dual Tone Multi-Frequency (DTMF) standard.

The tone indicator 57 also may represent other tones associated with a telephone call. For example, in response to a user lifting a handset from a switch hook or pressing a button on a telephone base, the tone indicator 57 may represent a dial tone to be generated by the tone generator 69. In response to attempting to establish a telephone call with another TCD that is busy, the tone indicator 57 may represent a busy signal and the tone generator 69 may generate, as the indicator tone 71, a corresponding busy signal. Further, in response to a call set-up message from anther TCD, the tone indicator 57 may represent an incoming telephone call indicator, for example, a ring. Other tones associated with a telephone call may be generated by the tone generator 69.

A first mixer 74 may receive the indicated tone 71 and stored audio data 58. The stored audio data 58 may be a portion of an audio file or other form of audio data stored in an audio storage medium 60. For example, the audio data 58 may be streaming audio that is a portion of a voicemail message or a prerecorded sound. The audio storage medium 60 may be non-volatile memory on the TCD or another storage resource on the communications network on which the TCD resides. The stored audio data 58 may be sent to the first mixer 74 in accordance with a telephony application.

The first mixer 74 may be configured with one or more of a plurality of mixing parameters 76. For example, one or more of the mixing parameters 76 may indicate weights to be associated with the indicated tone 71 and the stored audio data 58. These weights may weigh the amplitudes of the indicated tone 71 and stored audio data 58 to generate the mixed audio 75. Accordingly, the mixing parameters 76 may effectively disable mixing of either the indicator tone 71 or the stored audio data 58, may give either input greater impact in generating mixed audio 75 or may be weighted such that the indicator tone 71 in the stored audio data 58 are mixed equally.

The splitter 73 receives the mixed audio 75 and sends the mixed audio 75 to the second mixer 65 and a third mixer 83.

The second mixer 65 receives the mixed audio 75 and the de-echoed audio data 63 and mixes the two inputs to produce mixed user audio 67. This mixed user audio 67 is sent to the local call bridge 51, which determines the media sent to each of the participants in the telephone call. Accordingly, the second mixer 65 may be configured

with one or more of the plurality of mixing parameters 76 to weigh the amplitudes of the mixed audio 75 and the de-echoed audio data 63, thereby defining the impacts of each of these inputs on the mixed user audio 67.

The local call bridge 51 receives the mixed user audio 67 and one or more decoded connection media 49, each decoded network media 49 corresponding to a connection, and produces one or more connection mixed media 53 and a user-specific mixed media 81. The local call bridge 51 determines which of the decoded connection media 49 and the mixed user audio 67 to mix to produce the user-specific mixed media 81 and each of the one or more connection mixed media 53. Further, the local call bridge 51 may receive bridge parameters 52 that configure the local call bridge 51 for such mixing.

For example, the local call bridge 51 may be configured such that the user-specified mixed media 81 does not include some or all of the mixed user audio 67. Such a configuration prevents the user of the TCD from hearing the microphone audio data 55 spoken by the user. This prevention may be desirable not only because the user may not need to hear her own voice, but also because hearing her own voice with a slight delay as part of the mixed audio output 85 may confuse the user, or at least make the mixed audio output more difficult to understand.

Similarly, the local call bridge 51 may be configured to prevent each connection mixed media 53 corresponding to a first connection from including media of the connection input media 43 corresponding to the first network connection. For example, consider a conference call including participants U1 corresponding to the first TCD, a second user U2 corresponding to a second TCD and a third user U3 corresponding to a third TCD. The local bridge 51 receives a mixed user audio 67, M1, corresponding to the first user and two decoded connection media 59, M2 and M3, corresponding to the first user U1 and the second user U2, respectively. The local bridge 51 may be configured to mix media M1 and media M3 to produce a connection mixed media 53, C1, to be transmitted to the second user U2. Analogously, the local call bridge 51 may be configured to mix media M1 and M2 to produce connection mixed media 53, C2, to be transmitted to the third user U3.

Although some mixing configurations have been described above, the local call bridge 51 may be designed such that the bridge parameters 52 can configure the call bridge 51 to mix in any of a variety of ways. For example, as is described below in

relation to feature telephony applications, an application may define a particular mix, or a user may provide values through an application for one or more of the bridge parameters 52 that define a particular mix such that a particular combination of the mixed user audio 67 and the one or more decoded network media 49 may be mixed and prevented from being mixed in generating the user-specific mixed media 81 and/or the one or more connection mixed media 53.

Each connection media output interface 87 may include an encoder 89 that receives the connection mixed media 53 and encodes this media using known techniques to produce the encoded media 91. An output buffer 97 receives the encoded media 91 and configures the encoded media 91, for example, by adding transport protocol data (e.g., RTP and RTCP) to produce the connection output media 99. The connection output media 99 is transmitted to the TCD connected by the connection corresponding to the connection output media 99.

The third mixer 83 receives the user-specific mixed media 81 and the mixed audio 67, and mixes these inputs to produce the mixed audio output 85. The mixed audio output 85 may be sent to any of a variety of audio output devices such as, for example, a speaker of a telephone handset of the first TCD or a base speaker on the base of first TCD.

The recording buffer 93 may receive the mixed audio output 85 and store the mixed audio output to an audio storage medium 62 such as, for example, a non-volatile storage medium or another storage resource external to the first TCD, located on the communications network. The audio storage medium 62 and the audio storage medium 60 may be a same storage medium or may each be part of a same storage medium.

The third mixer 83 may be configured using one or more of the mixing parameters 76 to weight the amplitudes of the user-specific mixed media 81 and the mixed audio 75, thereby defining the impacts of each of these inputs on the mixed audio output 85.

Each of the processing elements of the media processing module 29, including 41, 42, 44, 47, 51, 59, 61, 65, 69, 73, 74, 82, 83, 89, 93 and 97, and data elements associated therewith, including 40, 43, 45, 49, 52, 53, 55, 57, 58, 63, 67, 71, 75, 76, 81, 85, 91 and 99, may be represented using software, firmware, hardware, or any combination thereof.

In an example software embodiment, each processing element and associated data elements may be represented using a general purpose object-oriented programming language such as, for example, C++ or Java.

Fig. 4 is a block diagram illustrating in an example embodiment of the call processing module 31 of Fig. 2 in relation to an example software embodiment of the media processing module 29. In this software embodiment of the media processing module 29, the media processing component 30 is represented using an object-oriented programming language such as C++ or Java.

The media processing module 29 may include a plurality of media processing controllers 115 and a plurality of media processing components 30, where each controller 115 and component 30 corresponds to a particular telephone call.

The media processing component 30 includes a plurality of media processing abstractions 117, where each media processing abstraction 117 represents a processing element from Fig. 3 and the input and output data elements associated with the processing element. Each media processing abstraction 117 may be of a different type, where each type inherits from a master media processing abstraction.

Each media processing abstraction 117 may include an attribute identifying the media processing component 30 to which the media processing abstraction 117 belongs and may include a plurality of methods that may be invoked on the media processing abstraction 117. The methods defined for one or more of the media processing abstractions 117 may include methods that do the following: enable and disable the media processing abstraction 117, initiate processing by the media processing abstraction 117 of the chunk of media it received, determine one or more states of the media processing abstraction 117, determine the media processing component 30 to which the media processing abstraction 117 belongs, determine a name or unique identifier of the media processing abstraction 117, determine information regarding the data input and output from the media processing abstraction, determine a maximum and minimum number of inputs and outputs for the media processing abstraction 117, determine the current number of inputs and outputs defined for the media processing abstraction, determine whether an input or an output is currently connected for the media processing abstraction 117, etc.

1.4.2.1.2 Media Processing Controller

The media processing controller 115 controls the creation, operation and destruction of the media processing component 30. More specifically, the media processing controller 115 controls the creation of the media processing abstractions 117, the definition of the relationships between the media processing abstractions 117, the operation of the media processing abstractions 117, and the destruction of the media processing abstractions 117.

For example, the media processing controller 115 may include methods for creating and destroying a media processing abstraction 117. The media processing controller 115 also may include a method for linking two media processing abstractions 117 together, thereby defining that the output of a first media processing abstraction is the input of a second media processing abstraction. The media processing controller 114 also may include a method for destroying such links, thereby destroying the relationship between two media processing abstractions. This adding and destroying of abstractions and links may be considered as building and removing, respectively, a media processing component 30 as part of setting up and tearing down, respectively, a telephone call.

Telephony applications of the applications layer 3, and other higher-level abstractions, may use the media processing controller 115 to dynamically create media processing components 30, and to link media processing elements during a telephone call. Further, a telephony application may be defined to configure media processing elements in unique ways to create custom media processing abstractions for a particular feature provided by the telephony application.

The media processing controller 115 may include methods for enabling and disabling a media processing component 30. If there are currently multiple media processing components 30 representing multiple telephone calls, it may be desirable to disable a particular media processing component 30 when the telephone call that it represents is not active (e.g., on hold) such that media processing resources are conserved. Conversely, when a telephone call becomes active, the enabling method may be used to enable the corresponding processing component 30.

The media processing controller 115 may include a method for indicating when a media processing component 30 is ready for processing media, and also a method for indicating that the media processing component 30 is no longer needed and may be destroyed.

The media processing controller 115 may include methods for initiating processing of a next chunk of data by the media processing component 30, locating a media processing abstraction 117, determining a number of media processing abstractions 117 of the media processing component 30 and determining a number of links and a media processing component 30.

As described above in relation to Fig. 3, a media processing component 30 may include representations of one or more connections. Accordingly, the media processing controller 115 may include an attribute corresponding to each connection represented by the media processing component 30. Further, the media processing component 30 also may include a method for retrieving a pointer to a connection.

As described above in relation to the connection media input interface 41 and the connection media output interface 87 of Fig. 3, media may be received and transmitted in the form of data packets conforming to one or more media transport protocols. Accordingly, the media processing module 29 may include an abstraction for representing a session, e.g., an RTP session, of one of the media transport protocols, and the media processing controller 115 may contain an attribute pointing to such attribute. Further, for each media processing abstraction 117 representing a connection of a telephone call, a media transport protocol such as, for example, RTCP, may maintain statistical information pertaining to the connection. The statistical information for the one or more connections may be aggregated and represented by an abstraction so as to monitor the transporting of the media on each connection. For example, this abstraction may be an RTCP session. The media processing controller 115 may contain an attribute that points to this RTCP session and may include a method for retrieving a pointer to the RTCP session.

For setting up a call between the first TCD and another TCD, the media processing controller 115 may include a method for determining one or more audio processing encoding algorithms supported by each TCD. As a result of this determination, the media processing controller 115 may control the decoding and encoding algorithms used by the decoder 47 and the encoder 89, respectively, of the media processing component 30.

The media processing controller 115 also may include a method for creating a connection media processing abstraction 117 such as, for example, the connection media input interface 41 and the connection media output interface 87.

The media processing controller 115 also may include methods for controlling the starting and stopping of media input to the media processing component 29. For example, the media processing module 29 may include methods that perform one or more of the following functions: controlling the tone generator 69 to start or stop generating the indicated tone 71; controlling the connection input media interface 41, in particular the input buffer 42, to start and stop generating the connection media 43; controlling the echo suppressor/cancellor 61 to start and stop receiving the microphone audio data 55; controlling the first mixer 74 to stop, start or pause receiving the stored audio data 58 to generate the mixed audio 75.

The media processing controller 115 also may include methods for controlling media output from the media processing component 30. For example, the media processing component may include methods to perform the following functions: controlling the user media output interface to start and stop sending the mixed audio output 85 to audio output hardware and to start and stop storing the mixed audio output 85 to an audio storage medium 62; and controlling the connection media output interface 87, in particular the output buffer 97, to start and stop outputting connection output media 99 to the communications network.

Although the first TCD currently may be involved in multiple telephone calls, the user of the first TCD can only actively participate in one telephone call at a time. For example, the user may have two telephone calls on hold, and be actively participating in a third telephone call. Because the user can only participate in one telephone call at a time, at any given time, only one media processing component 30 needs to use the one or more microphones and the one or more speakers of the first telephony communications device. Accordingly, the media processing controller 115 may include methods for assigning and de-assigning the microphones and speakers to a media processing component 30.

All of the functions described in relation to the media processing controller 115 and the media processing component 30 may be invoked in response to functions defined by higher-layer abstractions such as the telephony applications of the applications layer 3. Specifically, these telephony applications communicate with the call processing module 31 to control the call processing associated with one or more telephone calls, and the call processing module 31 controls the media processing module 29 to process media in accordance with the one or more telephone calls.

Representing the media processing and call processing of a call independently, e.g., with the call processing module 31 and the media processing 29, provides flexibility in designing the call processing module 31. For example, the media processing module may be implemented as part of a DSP or as one or more software abstractions. As a result, the call processing module 31 may be configured such that the abstractions of the call processing module 31 are generic to more than one implementation of the media processing module 29, such that less programming is needed to adapt the call processing module 31 to a particular implementation of the media processing module 29.

1.4.2.2 Call Processing Module

Fig. 4 shows the call processing module 31 of Fig. 2 in more detail. The call processing module 31 may model the state transitions of one or more telephone calls. Accordingly, applications running on the first TCD, for example, applications defined in the applications layer 3, may query the abstractions of the call processing module 31 to determine the state of a telephone call and its connections. Further, applications may register to be automatically notified if the state of a telephone call, including one of its connections, changes, or if other telephony events occur.

The call processing module 31 may include the following abstractions: a call manager 105, one or more calls 109, one or more call processing media interfaces 113, one or more call control connections 111, a call control transport controller 107, one or more call control network input interfaces 101 and one or more call control network output interfaces 103.

The processing performed by the call processing module 31 includes controlling one or more calls in accordance with one or more call control protocols, (e.g., signaling protocols) and controlling the media processing associated with each call and the one or more connections included in the call.

1.4.2.2.1 Call Manager

The call manager 105 may be considered the nucleus of the call processing module 31 and serves as an interface between the call processing module and higher layers of abstraction. The call manager 105 controls the one or more calls 109 that represent the telephone calls in which the first TCD is currently participating. The call manager 105 communicates telephony events to the appropriate call 109. Such events may include call control events, for example, a call control message, and application-

level events, for example, an event indicating to create a new call 109 or connection or to terminate a connection or call. For example, during a telephone call, or in setting up or tearing down a telephone call, higher level telephony applications may invoke methods defined by the call manager 105, which, subsequently, may invoke a corresponding method of the appropriate call 109.

Each call 109 may contain several methods, many of which correspond to the methods of the call manager 109, that are executed in accordance with the performance of a telephone call. It may be desirable to configure higher-level abstractions to communicate with the call manager 105, as opposed to an appropriate call 109, because calls 109 are relatively transient abstractions that may be available at one instance (e.g., during a telephone call) and gone at a next instance (e.g., after someone hangs up a telephone). Accordingly, the call manager 105 provides a less transient (i.e., more persistent) abstraction that is available regardless of whether the first TCD is currently participating in any telephony calls.

Because the call manager 105 and each of the calls 109 that it manages include several analogous methods, it should be understood that, for illustrative purposes, several methods are described below in relation to the call manager 105 or a call 109, but such methods may be applicable to both types of abstractions in accordance with how the abstractions are configured.

The call manager 105 may include several functions for controlling the setting up, maintaining, tearing down, and media processing of a telephone call.

The call manager 105 may include an attribute that maintains a list of all calls 109 currently represented by the call processing module 31, and may include methods for creating a call and dropping a call. For example, the call manager may create a call in response to a user dialing a number or entering a URL, or in response to the user answering a call by picking up a handset or pressing a button.

The call manager 105 may drop a call in response to a user of the first TCD hanging up or in response to a signal received from another TCD indicating that the user of that device has terminated the telephone call.

The call manager 105 may include a method for determining the number of calls 109 currently represented by the call processing module 31, and for determining the call states of each of these calls 109. The call manager 105 and calls 109 define states that model a telephone call similar to how the Java Telephony Application Programming

Interface (JTAPI) models a telephone call. Accordingly, the states defined for a telephone call may include: the call state, such as, for example, idle, active or invalid; the terminal state which represents the state of the first TCD; the address state that represents the first TCD's address (i.e., telephone number or URL); the call connection state that associates the first TCD's address with a telephone call; and the terminal connection state that associates the first TCD with a connection.

The terminal connection may be desirable to define a state of a particular terminal connection because one or more TCDs may be associated with a same address. For example, a home may have a plurality of telephones reachable at a same telephone number.

The call manager 105 may include a variety of methods commonly associated with a telephone call, including methods for adding a connection to a telephone call, dropping a telephone call, and transferring a telephone call to another TCD. Other methods may define, in response to receiving a call set-up message from another TCD, a function for accepting a connection with another TCD. Accepting a call may include invoking a function on the media processing interface 113 that controls the phone to ring. Further methods may define, in response to receiving a call set-up message, a function for either rejecting a connection, including invoking methods of the media processing controller 115 to control generating a busy signal and sending it to the calling TCD, or re-directing the calling TCD to another TCD.

The call manager 105 also may include a method for dropping a connection from a telephone call, for example, in response to input from the user of the first TCD or in response to receiving a signal indicating that the user of another TCD has hung-up.

The call manager 105 also may include methods for determining the number of call control connections 111 of a call 109, for getting control of a call control connection 111, and determining a state of a call control connection 111.

The call manager 105 also may include methods for determining addresses that have been called as part of a call 109 and for determining the addresses of the TCDs that have called the first TCD as part of the call 109.

The call manager 105 also may include methods for completing the set-up of a connection between the first TCD and a second TCD, for example, by sending a response to a call control set-up message or by sending an acknowledgement to another TCD acknowledging that the other TCD accepted the invitation to have a telephone call.

Such methods may communicate with the call control transport controller 107, described in more detail below.

1.4.2.2.2 Media Processing Interface

For each call 109, the media processing interface 113 provides an interface between the call 109 and the media processing controller 115 and media processing component 30 corresponding to the call 109. Through the media interface 113, a call 109 may control its own media processing independently from the media processing of other calls 109.

The media processing interface 113 may include several methods corresponding to the methods of the media processing controller 115. Further, the media processing interface 113 may include methods that control the media processing module 29 to process media in accordance with a specific telephony event. As described above in relation to Fig. 4, the media processing controller 115 may include functions for controlling the tone generator 69 to start and stop the generating of indicated tones 71. The media processing interface 113 may include functions to control the media processing controller 115 to control generation of a specific tone corresponding to a telephony event. For example, the media processing interface 113 may include functions for starting and stopping the playing of specific tones such as DTMF tones, dial tones, a tone indicating that another TCD is busy (e.g., a busy signal), and a tone indicating that a telephone call is incoming (e.g., a ring), etc.

The media processing interface 113 may function as described in the following example. The call manager 105 may receive an indication, e.g., from the phone set management module 25, that a user has pressed a key on a telephone keypad. The call manager 105 determines the call 109 associated with the pressed key and invokes a method on the call 109 to play the DTMF tone associated with the key. Consequently, the call 109 invokes the appropriate method on the media processing interface 113 to start playing the DTMF tone, and this method invokes the appropriate method of the media processing controller 115 to control the tone generator 69 to receive the tone indicator 57, which indicates the DTMF tone, and to generate the DTMF tone as the indicated tone 71.

The media processing interface 113 also may include one or more methods for querying the media processing module 29. For example, the media processing interface

113 may include methods for determining that the media processing module is receiving connection input media 40 and transmitting connection output media 99.

The media processing interface 113 also may include methods for querying the phone set management module 25 regarding the status of low-level devices, such as the hook switch. For example, the media processing interface 113 may include one or more methods for querying one or more abstractions of the phone set management module 25 to determine if a handset is off-hook.

The media processing interface 113 also may include methods for determining the codec algorithms supported by the TCD, for assigning an audio output device to send the mixed audio output 85, and for determining whether an audio output device is currently enabled, for example, by querying the phone set management module 25.

The media processing interface 113 may be any of a variety of types of media processing interfaces 113. For example, as part of the telephony system architecture 1 residing on the first TCD, the media processing interface 113 is of a type defined specifically for a TCD. In another example, the call processing module 31 may model the call processing performed on a media gateway between two networks or a signaling gateway between different signaling protocols. In these cases, the media processing interface 113 may be of a type that is defined specifically for the type of processing necessary for the particular gateway.

The media processing interface 113 deals primarily with processing media. Another aspect of the call processing module 31 is to control a telephone call, including the use of call control (i.e., signaling) protocols. This aspect of the call processing module 31 will now be described in more detail.

1.4.2.2.2 Call Control Protocols

A call 109 may be of a particular species of call, for example, a peer-to-peer (peer call) or a master/slave call (slave call). A peer call 109 represents a call that includes one or more connections adhering to a peer-to-peer protocol, for example, SIP or H.323, and may include one or more connections adhering to a master/slave protocol, for example, MGCP, Megaco/H.248 or Skinny Station. A slave call 109 represents a call that includes one or more connections adhering to a master/slave protocol. Each species of call 109 includes abstractions for defining data corresponding to the type of connections included in the call and for defining methods for controlling a telephone call in accordance with the species of protocol.

A slave call 109 is a call that is controlled by a network resource external to the first TCD. A slave call 109 may be one of a plurality of master/slave call types, including an MGCP call, a Megaco/H.248 call or a Skinny Station call. MGCP calls 109, Megaco/H.248 calls 109 and Skinny Station calls 109 are calls that adhere to MGCP, Megaco/H.248, and the Skinny Station protocol, respectively. Each of these types of slave call may be defined as an abstraction that inherits from an abstraction of a generic slave call 109, and may further define functionality associated with the type of slave call that it represents.

The call processing module 31 includes one or more call control connections 111. Each call control connection 111 (except a ghost call connection described below in more detail) has a corresponding media control call connection as represented by a media processing abstraction 117 of the media processing module 29. As described above in relation to the media processing component 30, such media processing abstractions represent the media flow of the connection. In contrast, a call control connection 111 of a call 109 defines and controls the signaling communications on the first TCD that are associated with a call 109 in accordance with a particular call control protocol.

A slave call 109 may include a plurality of slave call control connections 111 that control the call signaling communications for the call on the first TCD in accordance with the type of master/slave call control protocol that the slave call control connection 111 represents.

For a peer call 109, each call control connection 111 may be one of several types, including an H.323 call control connection, a SIP call control connection, a ghost call control connection and a slave proxy call control connection.

An H.323 call control connection and a SIP call control connection control the signaling communications of a connection on the first TCD in accordance with the H.323 protocol and the SIP protocol, respectively.

A ghost call control connection represents a connection between two other TCDs. For example, if the first TCD transfers a call to another TCD and for some reason it is desired to track the progress of the transferred connection, a ghost call control connection may be used to monitor the state of the connection between the two other TCDs. To monitor the state of the transferred connection, the ghost call control connection may receive communications from one of the two other TCDs. Because the connection is

between the two other TCDs, a ghost call control connection does not have a corresponding media processing connection.

If a peer call 109 represents a call including a slave connector, a slave proxy call control connection 111 that proxies a slave call control connection 111 may be used to represent the slave connection 111.

A telephone call represented by a call 109 may include several connections, each connection represented by a call control connection 111. As described above, each connection included in the telephone call may adhere to any of a plurality of call control protocols. Accordingly, a call 109 may control one or more call control connections 111, where each call control connection 111 is of a different type, for example, an H.323 call control connection or a SIP call control connection.

Representing each connection of a telephone call with an independent call control connection 111, and independently representing the media processing and call processing of a telephone call enables a call 109 including multiple connections 111 to represent and control each connection concurrently, where each connection may adhere to any of a plurality of call control protocols.

For example, during a telephone call between the first TCD, a second TCD and a third TCD, a first connection may exist between the first and second TCDs, and a second connection may exist between the first and third TCDs. The first connection may be represented by a SIP call control connection 111, and the second connection may be represented by an H.323 call control connection 111. Further, the first and second connections may be represented by independent media processing abstractions 117 that represent the media processing of the connection.

The call 109 that represents such a telephone call may control the SIP and H.323 call control connections 111 and the corresponding media processing abstractions 117. During the telephone call, for each connection 111, in accordance with the call control protocol that the connection represents, SIP or H.323, the call 109 may invoke methods of the media processing interface 113, which invokes methods of the media processing controller to control the processing of media on the appropriate connections through the corresponding media processing abstractions 117.

The call control connections 111 control the signaling communications of a connection on the first TCD at a relatively high level. The call processing module 31 also includes a call control transport controller 107 that controls the signaling

communications of a connection on the first TCD at a lower level that controls the transport and ensures the reliability of signaling messages transmitted from and received at the call processing module 31.

The call processing module 31 also includes call control network input interface 101 for receiving call control messages corresponding to a call control protocol, and a call control network output interface 103 that transmits call control messages from the call control transport controller 107 to the communications network.

To send a call control message, for example, a SIP message, a call control connection 111 sends the message to the call control transport controller 107 which then sends the call control message to the call control network output interface 103, which then sends the call control message on to the communications network.

For incoming call control messages, the call control network input interface 101 receives the call control message and communicates the message to the call control transport controller 107, which communicates the incoming call control message to the call manager 105. The call manager 105 then dispatches the call control message to the appropriate call 109 that includes the connection 111 representing the network connection from which the call control message was received.

The call control transport controller 107 also notifies the call manager 105 of sent call control messages that have been sent on to the communications network, but for any of a variety of reasons failed to reach their destination. Further, in response to such a failure, the call control transport controller 107 may re-send a call control message onto the communications network through the call control network output interface 103.

Using the various abstractions described above, the call processing module 31 may be configured to support any of a plurality of call control protocols. Further, the call processing module 31, or an application using the call processing module 31, may be configured to select one of the plurality of call control protocols for a particular connection. This selection may depend on any of a number of factors, including available network resources and the capabilities of the TCD connected to the first TCD by the connection.

For outgoing calls from the first TCD, the call processing module 31 or another telephony application may be configured to select a call control protocol to set-up a call with another TCD based on a sequence of digits or a URL entry by a user. For example, an application may define a rule indicating that, if a user enters a three-digit extension,

the H.323 protocol is used to control the telephone call. Further, an application may include a rule indicating that, if a user enters a ten-digit PSTN telephone number, then SIP is used to control the telephone call.

In yet another example, an application may define a rule indicating that, if a user enters a URL including the character string "SIP", then the SIP protocol is used to control the telephone call, and if the user enters a URL including the character string "H.323", then the H.323 protocol is used.

Significantly, because the details pertaining to different call control protocols are handled by the call processing module 31, the core telephony API layer 21 and open API layer 5, each described in more detail below, may be used to develop applications that are generic to all the call control protocols, allowing the call control details to be handled by the call processing module 31. Thus, call processing module 31 allows API's of both the core telephony API layer 21 and open API layer 5 to be call-control-protocol-independent.

Although the first TCD may be configured to support any number of call control protocols, and to select a particular call control protocol for a connection, the call processing module 31, another module of the core telephony functionality layer 9, or one or more telephony applications of the applications layer 3 may configure the first TCD to support only certain call control protocols. For illustrative purposes, the various techniques that may be used to configure the first TCD to support particular call control protocols will be described with reference to the call processing module 31, although other modules or layers may define applications to apply the same techniques.

The call processing module 31 may be configured to determine the call control protocols supported by the communications network on which the first TCD resides. For example, upon being initialized for the first time at a customer's premises, the call processing module 31 may send communications onto the customer's communication network to determine what call control protocols are supported, to determine any network resources that the first TCD should communicate with to implement the call control protocol, and to load any software necessary to support the call control protocol. Further, the call processing module 31 may be configured to determine which call control protocols are supported by the communications network for incoming and outgoing telephone calls.

The call processing module 31 may be configured to perform the discovering process in any of a number of ways, including: as a result of being initialized in the field, periodically, upon request by a user, or any combination thereof. Further, this discovery process may be performed for the first TCD by another network resource, for example, a TCD deployment server, discussed below in more detail in connection to Fig. 7.

In an embodiment, the call processing module 31 is programmed to know the location of the one or more network resources that may be used for implementing one or more call control protocols.

A communications network on which the first TCD resides may include a Dynamic Host Configuration Protocol (DHCP) server. The DHCP server may be configured to store information about the call control protocols that are supported by the communications network, and may indicate one or more servers that may be used to help implement the one or more call control protocols. If such a DHCP server is present, the call processing module 31 may access the DHCP server to retrieve the above-described call control information. Using DHCP servers to discover call control servers is described in the Internet draft document by the IETF entitled "DHCP Options for Call Control Servers" located at URL: http://www.ietf.org/internet-drafts/draft-ietf-dhcp-01.txt as of April 6, 2000.

If the communications network does not include a DHCP server, or for some reason the DHCP server does not return the necessary call control protocol information, the call processing module 31 may be configured to communicate with one or more other network resources to determine the call control protocols, and their associated servers, supported by the communications network.

For example, to determine whether the communications network supports the H.323 protocol, and the location of any H.323/SIP signaling gateways on the communications network, gatekeeper discovery techniques may be used. For example, the gatekeeper discovery techniques described by the International Telecommunications Union in the ITU-T Recommendation: H.323 Packet-Based Multi-Media Communications Systems, Geneva, Switzerland, February 1998.

To determine if the communications network supports the SIP protocol, and the location of any SIP servers on the communications network, the call processing module 31 may be configured to locate the one or more SIP servers using SRV DNS records as

outlined in Appendix of RFC 2543, SIP: Session Initiation Protocol by IETF as of October 26, 1999. If this SIP discovery technique fails, the call processing module 31 may be configured to attempt to contact a SIP proxy server at *sip.<local domain>* at default port 5060, using UDP and/or TCP.

To determine whether the communications network supports the MGCP protocol, and the location of any MGCP call agents, the call processing module 31 may be configured to use the SRV DNS records approach (analogous to the technique described above for SIP in RFC 2543). An MGCP call agent is an entity responsible for controlling telephone calls for a communications network using the MGCP protocol. The call processing module 31 may be configured such that, if this attempt to locate the MGCP call agent is unsuccessful, an attempt is made to contact an MGCP call agent at *mgcp.<local domain>* using UDP at the default port for MGCP, port 2427.

To determine whether the communications network supports the Megaco/H.248 protocol, the call processing module 31 may be configured to use a similar technique to that described above for the MGCP protocol. The Megaco/H.248 protocol standard is defined in IETF RFC 2885 and ITU H.248.

The call processing module 31 may be configured such that if the communications network does not support a particular call control protocol, then a space-saving stub may be installed as the call control connection 111 for the particular call control protocol.

The call processing module 31 may be configured to use a particular call control protocol, for example, SIP, as the call control protocol for all outgoing calls. If, however, the call processing module 31 determines that only a single call control protocol is supported by the communications network, the call processing module 31 may assign this call control protocol as the protocol used for all outgoing calls.

Further, if the call processing module 31 determines that the communications network supports multiple call control protocols for outgoing calls, but only a single call control protocol for inbound calls, the call processing module 31 may assign this single call control protocol as the call control protocol used for all outgoing calls.

The call processing module 31 also may be configured to use a particular call control protocol as the call control protocol for all incoming calls. If, however, the call processing module 31 determines that only a single call control protocol is supported by

the communications network, call processing module 31 may assign this call control protocol as the protocol used for all incoming calls.

Further, if the call processing module 31 determines that the communications network supports multiple call control protocols for incoming calls, but only a single call control protocol for outbound calls, the call processing module 31 may assign this single call control protocol as the call control protocol for all incoming calls.

By using abstractions 101-113, the call processing module 31 may be configured to support several features in accordance with a call control protocol e.g., SIP. Such call control features may include: transmitting and receiving call control messages in accordance with a variety of transport layer protocols, for example, Transport Control Protocol (TCP) and User Data Protocol (UDP); basic call set-up functions such as sending a call set-up message and responding to a call set-up message; and firewall support.

The call processing module 31 may support registering the first TCD in a registry or directory, for example, on a SIP server. Such a directory may reside on the first TCD, or on another network resource, for example, a companion computer to the first TCD or a network server. As described in more detail below in relation to Fig. 7, each entry of such a server may map a name or a URL of a user to a network address of the first TCD.

The call processing module 31 also may support having a registry entry corresponding to the first TCD expire after a predetermined amount of time. Further, the module 31 may be configured such that such entry is refreshed periodically to avoid expiration of a user's entry while the TCD is active (i.e., powered on).

The call processing module 31 also may be configured to initiate placing a telephone call on hold. For example, a signal may be sent from the first TCD to another TCD indicating that the first TCD will stop consuming bandwidth. Analogously, the call processing module 31 also may be configured to receive a message from another TCD indicating that the other TCD will stop consuming bandwidth (i.e., is placing the telephone call on hold).

The call processing module 31 may be configured to redirect phone calls in a plurality of ways. For example, the call processing module 31 may be configured such that if the user of the first TCD is already participating in another phone call, an incoming telephone call is redirected to another device on the communications network. For example, the other device may be a voicemail server that plays the user's voicemail,

or may be another TCD. Further, user input, for example, from a button on a user interface may indicate that the user does not want to be disturbed regardless of whether the user is participating in a telephone call. Accordingly, an incoming call will be redirected automatically. Alternatively, an incoming call may be redirected if the incoming call is not answered after a predetermined amount of time or after a predetermined number of rings. Lastly, an incoming call may be redirected on demand in response to receiving a user input.

The call processing module 31 may be configured to support blind transfers and consultative transfers of a telephone call. To perform a blind transfer, the call processing module 31 may be configured, for a telephone call between a first user of the first TCD and a second user of a second TCD, to instruct the second TCD to set-up a call with a third user of TCD. After instructing the second TCD, the first TCD will terminate the connection between the first TCD and the second TCD without first consulting with a user of the third TCD.

For a consultative transfer, the first TCD will set-up a connection to the third TCD. The first user may then consult with (i.e., talk to) a user of the third TCD, after which the first TCD may instruct the second TCD to set-up a call with the third TCD, or may instruct the third TCD to establish a connection with the second TCD.

For either a blind transfer or a consultative transfer, an application may monitor the progress of establishing the new connection using a call 109 including a ghost call control connection 111. The other TCD that was instructed by the first TCD to set-up the new telephone call that includes the new connection may send one or more call control messages to the first TCD to notify the first TCD of the status of the new connection. If for some reason, it is desired to monitor and/or record the status of the new connection, the ghost call control connection 111 may maintain this status information.

The call processing module 31 also may be configured to record routing information regarding one or more connections associated with the telephone call. Such routing information may include the duration of a connection as part of a telephone call, and this routing information may be used for several purposes including billing.

The call processing module 31, in accordance with the call control protocol of a connection, also may support attaching to a call control protocol message non-call control protocol information, for example, audio files, text files, applications, data files,

and pointers to data files or applications, etc. For example, the call processing module 31 may be configured such that multi-part MIMEs (Multi-purpose Internet Mail Extensions) may be attached to a SIP message.

The call processing module 31 may be configured, in accordance with a call control protocol of a connection, to inform another TCD of the capabilities and/or additional features and functions supported by the first TCD. For example, the call processing module 31 may be configured to use fields of a SIP message to inform the other TCD of the capabilities and additional features of the first TCD, and may negotiate common capabilities to be used between the two TCDs.

The call processing module 31 may be configured to record the duration of a connection between the first TCD and another TCD. Further, call processing module 31 may be configured to send and/or receive a message to/from another TCD at pre-determined intervals to indicate that the sender of the message is still participating in the telephone call. Sending such messages at a pre-determined interval prevents an error in determining the duration of a phone call when a call control message to tear down a telephone call is never received.

The call processing module 31 may be configured to support several different known methods of authentication, including basic and digest methods of authentication. Further, the call processing module 31 may be configured to provide different known authentication on the first TCD including symmetric authentication along with another TCD, and proxy authentication, where an authentication is performed for another network resource.

Further, the call processing module 31 may be configured to re-invite a call participant back into a telephone call. Re-inviting may be desirable to change the encoding algorithm used for a connection in response to changes in available network bandwidth or in response to a change in the number of connections involved in a conference call.

The call processing module 31 may be configured to combine two telephone calls into a single telephone call. For example, call manager 105 may manage a first telephone call 109 including a first SIP call control connection 111 and a first H.323 call connection 111, and a second telephony call 109, including a second SIP connection 111, where each call control connection 111 also has a corresponding media processing connection 117. To combine the two telephone calls, the call manager 105 may be

configured to create a third connection on the first call 109 that represents the second SIP connection 111 and to create a corresponding media processing connection 117. After transferring all the information regarding the second call 109 and the second SIP connection 111, the call manager 105 may then drop the second call 109. As a result, the first call 109 remains, including two SIP call control connections 111, an H.323 call connection 111 and three media processing connections 117.

Returning to Fig. 2, the phone set management module 27, the media processing module 29 and the call processing module 31 are relatively low level modules including abstractions that capture telephony events and notify other abstractions of these events. The notified abstractions may be abstractions defined by one of the other modules 27, 29 or 31 within the core telephone functions module 25, or may be abstractions from the telephony application objects (TAO) module 23.

1.4.3 Telephone Application Objects Module

The TAO module 23 may include a plurality of telephone application abstractions such as, for example, a client abstraction, a server abstraction, a message abstraction, a transport abstraction to enable local and remote communications between telephony applications defined in the applications layer 3 and the core telephony functions defined by the core telephony function module 25 and abstractions corresponding to the telephony abstractions defined by JTAPI.

The TAO layer 23 may include logic enabling synchronous and asynchronous communications for handling request-response commands and event notifications, and abstractions defining interfaces to the core telephony functionality API layer 21 to enable remote function calls on abstractions of the core telephony functionality layer 21.

Thus, the TAO layer 23 allows layers 15, 13, 11, 33 and 25 to remain independent, logically and physically, from the remaining layers 21, 5 and 3. Accordingly, layers 21, 5 and 3 may be embodied entirely within the first TCD, may be embodied entirely on a network resource external to the first TCD and invoke remote function calls on the abstractions of the core telephony functionality layer 9 to develop and execute telephony applications or may be embodied partially within the TCD and partially external to the first TCD.

The TAO layer 23 may be programmed in an object-oriented programming language such as C++ or JAVA, and may include abstractions corresponding to the abstractions defined by JTAPI.

1.4.4 Core Telephony Functionality API

The core telephony functionality API 21 provides a programming interface to the core telephony functionality of the first TCD. The core telephony functionality API 21 may include several abstractions corresponding to the abstractions provided by JTAPI. For example, the core telephony functionality API 21 may include several abstractions written in C++ that were patterned after from the JAVA abstractions of JTAPI. The core telephony functionality API 21 may be implemented using the Pingtel Telephony API (PTAPI) available from Pingtel Corporation of Woburn, Massachusetts.

As described above, each abstraction of the core telephony functionality API 21 may have a well-defined state and a set of events that are triggered as a result of state transitions. Analogously to JTAPI, the core abstractions of the core telephony functionality API layer 21 may include abstractions for: a provider, an address, a terminal call, a connection, a terminal connection, a terminal and listeners. The core telephony functionality API layer 21 may be configured such that only privileged vendors have access to, and thus may manipulate, the core telephony functionality of the first TCD.

Telephony system architecture layers 9-15 provide the first TCD with core telephony functionality and a programming interface for privileged vendors to manipulate this core telephony functionality and add additional functionality to it. Thus, if the first TCD were deployed in the field with just layers 9-15, the telephony functionality of the first TCD could be expanded or modified only by vendors having privileged access.

Denying general access to certain core telephony functionality may be desirable so that the first TCD operates properly. For example, it may desirable to limit access to abstractions defined in the core telephony layer that controls structuring a call set-up message to be in conformity with a particular call control protocol, for example, SIP. Failure to limit such access may result in a user of third party vendor or user corrupting such abstractions such that the first TCD is incapable of sending messages onto the communications network in conformance with the SIP protocol.

On the other hand, it may be desirable to enable users and third party vendors to develop applications to be run on or in conjunction with the first TCD. Therefore, it may be desirable to have a telephony system architecture that is more open and extensible.

1.5 Open API Layer

Accordingly, the telephony system architecture 1 may include layers 1-5 to allow the telephony functionality of the first TCD to be expanded and modified after being deployed in the field.

The open API layer 5 includes abstractions that are the building blocks for applications developed for the application layer 3 by the API layer 5. For example, the application open API layer 5 may include telephony framework classes written in the JAVA programming language and may include a JAVA virtual machine (JVM®) from Sun Microsystems, Inc.

The open API layer 5 provides open APIs for software developers to develop applications, including telephony applications and non-telephony applications, for the applications layer 3. To make the open API layer 5 open to software developers, the APIs of the open API layer, for example APIs 6, 7 and 8, and 10 may be distributed or made publicly available on the Internet. For example, one or more of APIs may be provided by the xpressa Development Kit™ available from Pingtel Corporation at http://www.pingtel.com.

The open API layer 5 may include one or more user interface APIs 6, one or more system APIs 7, one or more telephony APIs 8, and one or more media APIs 10.

1.5.1 User Interface API

A user interface API 6 assists a developer in developing a graphical user interface (GUI) for the first TCD. The user interface API 6 may provide a plurality of controls and forms for designing the GUI. For example, the user interface API may be the JAVA abstract window tool kit (AWT) or the xpressa Window Toolkit™ (xWT), an extension of Java AWT, available from Pingtel Corporation at http://www.pingtel.com.

1.5.2 Telephony API

The telephony API 8 provides a plurality of abstractions for adding functionality to and for manipulating the functionality of the first TCD. The telephony API 6 may be JTAPI from Sun Microsystems, Inc., or an implementation thereof. The JTAPI specification is available at http://java.sun.com/products/jtapi/. Accordingly, the telephony API 6 may include a plurality of abstractions corresponding to JTAPI, including abstractions for: an address of the first TCD, telephone calls in which the first TCD is participating, connections included within these telephone calls, one or more

providers that provide telephony services corresponding to the first TCD, a terminal corresponding to the first TCD, etc. Other abstractions may include Observers, Listeners, Exceptions, events, states, methods associated therewith, or any other abstractions provided by JTAPI.

Other telephony APIs 8 may include simplified versions of JTAPI for more streamlined and easier use. For example, the telephony APIs 6 may include the simple telephony API (STAPI) available from Pingtel Corporation at http://www.pingtel.com.

In addition to event abstractions and listener interfaces, the telephony APIs may include hooks that allow application developers to affect certain time-critical behaviors of the first TCD. A hook is a method, for example, a JAVA method, that is invoked by an abstraction if an event of a specified type occurs. For example, each of the telephony APIs 6-8 may provide hooks that allow a developer to affect the processing of caller-ID information, the redirecting or filtering of phone calls, and determining how the first TCD indicates to a user that a call is incoming. For example, the first TCD may indicate that a call is incoming visually, for example, by blinking a light, audibly, for example, by beeping, or by any combination thereof.

The open API layer 5 also may include a language interface to interface abstractions defined by the open API layer 5 with abstractions defined by the core telephony functionality layer 9, in particular, abstractions defined by the TAO module 23. For example, if the one or more APIs of the open API layer 5 are defined in Java, and the TAO object module is defined in C++, the language interface may be the Java Native Interface (JNI).

1.5.3 System API

The system API 7 provides system-wide tools for the telephony system architecture 1. For example, the system API may include an application manager for starting, stopping, loading, unloading and monitoring applications running in the application layer 3. Further, the system API 7 may include an application arbitrator to regulate applications, prevent conflicts between applications, and to resolve conflicts between applications. The system API may provided as part of the xpressa Development Kit™.

1.5.3.1 Application Manager

The application manager may consist of a web server residing on the first TCD that enables the loading and unloading of applications onto the first TCD. Accordingly,

a user may use a web browser to access the web server. This web browser may reside on the first TCD itself, or on another network resource such as a computer. For example, as described below in relation to Fig. 7, if the first TCD is embodied as a telephone, it may be desirable to install the web browser, and other tools, on a computer to provide an applications developer or other person access to the computer's more extensive resources (e.g., memory, video monitor, keyboard, mouse) to develop and upload applications.

To add an application to the applications layer 3 through the application manager web server, a user may specify a location and file name. For example, through the web browser, the user may specify a URL at which the executable file, for example, a JAVA archive (.JAR) file, is located.

Depending upon the configuration of the first TCD, the application manager web server may store the specified file on the first TCD itself, store the specified file on another network resource and maintain a pointer to that location, or store a pointer to the URL specified by the user.

To uninstall an application, a user may merely enter the application name through the web browser, and the application manager web server will de-install the specified application.

The application manager may be implemented using other types of applications besides a web-based client/server application.

1.5.3.2 Application Arbitrator

The applications arbitrator may be configured to provide two general functions, security and conflict resolution.

The applications arbitrator may provide security for the first TCD by regulating the addition, modification and execution of applications. For example, the applications arbitrator may require an application being installed, or modified and re-installed, on the first TCD to meet certain criteria. For example, the application may have to register a list of first TCD resources, e.g., abstractions, that the application when executed is going use. The applications arbitrator may be configured to reject the application if the application does not include such a list.

If the application does include a list, after inspecting the list, the applications arbitrator may reject the application, requiring the application to be modified to not use certain resources. Alternatively, the applications arbitrator may accept an application,

and attach an indication of approval to the application, so that the application may be installed and executed on the first TCD.

The application arbitrator may be configured to accept or reject applications automatically, or may provide a user interface to allow a user, e.g., a systems administrator, to review the list of resources for each application and manually accept or reject the application.

The applications arbitrator also may be configured to monitor the application during execution. If the application uses a resource that it did not register with the application arbitrator, the application arbitrator may prevent the application from executing any further.

The other general function provided by the applications arbitrator is conflict resolution. On typical communications networks, conflicts may occur between telephony functions running on the communications network. Historically, PBX-based telephony networks have tested for possible conflicts and worked to provide appropriate behavior if a conflict occurs. On a communications network having one or more TCDs with modifiable and expandable functionality, however, each TCD may have a unique set of applications defined thereon and, consequently, a more diverse array of conflicts may occur. Therefore, it may be desirable to apply more proactive techniques for addressing these potential conflicts, including techniques to prevent conflicts between applications and techniques to identify and resolve conflicts if they do occur.

Accordingly, the application arbitrator may be configured to resolve conflicts between two or more applications residing or being executed on the TCD.

As discussed above, applications may communicate with abstractions of the core telephony functionality layer 9 to monitor telephony events. Specifically, applications may register to be notified as a result of an event occurring. To deal with conflicts between applications, first, for events for which two or more applications are registered to be notified, the application arbitrator may be configured to define an order in which the applications are to be notified. Second, the application arbitrator may be configured to define whether an application either a) handles and relays a telephony event, or b) consumes and does not relay a telephony event.

If an application handles an event, after the application executes functionality in accordance with the occurrence of the event, the application then notifies the next registered application of the occurrence of the event. If an application is defined to

consume an event, then after the application executes functionality in accordance with the occurrence of the event, the application does not notify any further registered applications of the occurrence of the event.

For example, a call waiting application may be configured to monitor a current call event defined by the call processing module 31 for any state changes, and may monitor incoming call events from the network to determine if a new call is incoming. If the state of the current call has not changed (i.e., the telephone call is still ongoing), then if an incoming call event occurs, the call waiting application may either (a) handle the event which triggers call waiting and also passes information about the new incoming call to any other applications that have registered for such notification; or (b) consume the event, which triggers call waiting, but does not pass any information about the new call to any other applications, even those applications that are registered for automatic notification. For example, even though a call forwarding application may be registered to be notified when an incoming call occurs, if the call forwarding application comes after the call waiting application, and the call waiting application is defined to consume the incoming call event, then the call forward application will not be notified of the incoming call or receive any information regarding the incoming call. By preventing the call forwarding application from being notified about the incoming call, a conflict between the call waiting application and the call forwarding application has been avoided.

1.5.4 Media API

Although the telephony API 8 enables a developer to develop a telephony application using well-defined relatively higher-level building blocks, e.g., abstractions provided by JTAPI, it may be desirable to design a more-specialized media processing application using lower-level media processing building blocks.

Accordingly, the media API 10 may provide tools to allow a developer to directly access and manipulate media abstractions of the media processing module 29, described in more detail above in relation to Figs. 3 and 4. A developer may develop a specialized media processing application to run on the first TCD by using the media API 10 to define custom media functions, for example, by invoking methods of the media processing controller 115 to link, create and destroy media processing elements.

For example, the media API 10 may be used to create and combine various media processing elements of the media processing module 29 to develop an application that

allows two or more participants in a conference call to conduct a side conference during the conference call, or an application that enables streaming audio conforming to a specific protocol, e.g., MP3, to be mixed into a telephone call, or an application implementing a customized voicemail system on the first TCD.

Telephony applications of the applications layer 3, and other higher-level abstractions, may use the media processing controller 115 to dynamically create media processing components 30, and to link media processing elements during a telephone call. Further, a telephony application may be defined to configure media processing elements in unique ways to create custom media processing abstractions for a particular feature provided by the telephony application.

2. Telephone Implementation of a Telephony Communication Device

Fig. 5 is a diagram illustrating an example embodiment of the first TCD as a telephone 200. The telephone 200 may be a desktop telephone having a connection to the network medium of the communications network. For example, the telephone 200 may be a desktop LAN-connected telephone such as the Pingtel xpressa™ by Pingtel Corporation of Woburn, Massachusetts, and may have an ornamental design as shown in U.S. Design Patent Application No. 29/120,479 entitled, "Telephone Base," by James A. Batson, Jr. et al., filed March 20, 2000.

Alternatively, the first TCD may be implemented in a more compact form, for example, a battery-operated wireless telephone such as an analog mobile telephone, a Personal Communications Service (PCS) wireless telephone, or a third generation (3G) wireless telephone.

As shown in Fig. 5, the telephone 200 provides a user interface, including a telephone base 202 and a handset 204. The telephone base 202 may include: a display screen 206; a scroll wheel 208; context-specific buttons 210; more button 212; dialing buttons 214; volume buttons 216; telephony function buttons 218; and speaker phone button 220 corresponding to base speaker/microphone 222, and visual indicator 224.

The dialing buttons 24 may be used for dialing telephone numbers. Alternatively, the telephone base 202 may include other interface controls, for example, a rotary dialer, for entering a telephone number.

Each of the telephony function buttons 218 may be used to perform a specific telephony function, for example, activating/de-activating the handset, transferring a call, muting the microphone base, conference call, and placing a call on hold. Further, each

button 218 may have an associated visual indicator that is lit while the button is active. For example, for a hold button 218, an indicator associated with the button 218 may be lit if a call is on hold.

The visual indicator 224 may be used to indicate one or more telephony events, for example, that a user of the telephone 200 has voice messages or that a call is incoming. Other visual indicators may be induced on the telephone 200.

The display screen 206 may be a liquid crystal display (LCD) and may display data and other information supplied by an application. Further, the display screen 206 may be designed with logic to be touch-sensitive such that selections and entries may be entered on the display screen 206 by touch. User interface controls 208, 210 and 212 may be used to access the displayed data and information.

The scroll wheel 208 may be integrated with the graphical user interface such that it may manipulate the position of information on the display screen 206, and may be used to select items displayed on the display screen 206. The scroll wheel 208 allows a user to interact with pages of content displayed on the display screen 206. A user may use the scroll wheel to change information displayed on the display screen 206 by scrolling or browsing down through the data, and then may select a specific item.

One or more applications may configure the scroll wheel to be application-dependent. Such applications may provide multiple options for both the scrolling and selecting activities so that applications may be configured with a desirable combination of these activities.

The direction in which the scroll wheel 28 is moved may determine forward or reverse progress through the content displayed on the display screen 206. For example, a clockwise rotation may cause forward movement through the content, and a counter clockwise rotation may cause reverse movement through the content, or vice versa.

A detent is a device that checks motion. The scroll wheel 208 may use detents to provide feedback to the user. Each momentary stop that occurs while the scroll wheel is turned indicates that a new position has been reached. Optionally, the scroll wheel 208 may be configured such that it may be turned continuously in either direction, each detent indicating a single forward or reverse movement, and the scrolling rate may be stable from detent to detent.

Alternatively, the scroll wheel 208 may be configured such that the extent to which the scroll wheel 28 may be turned is limited to a particular number, for example,

three, detent positions in each direction, and increasing resistance may be provided as the wheel is turned. Further, as a result of a user releasing the scroll wheel 208, it may automatically return to a detent representing a rest position between the sets of directional detents.

The scroll wheel 208 may be configured such that users may select a particular scroll rate using the detents. Alternatively, scroll rates may be fixed or may be a combination of fixed and variable. Table 1 below illustrates various rate options for given scroll wheel positions with which the scroll wheel 208 may be configured.

Scroll wheel position	Fixed rate option	Fixed and Variable rate option
rest - 1st detent	fixed, slow rate	fixed, slow rate
1st - 2nd detent	fixed, medium rate	fixed, medium rate
2nd – 3rd detent	fixed, fast rate	accelerating, fast rate

Table 1

For the combined fixed and variable rate option, an A-D converter may be used to sense the scroll wheel position.

The telephone 200 may be configured with a variety of GUI methods to select an item from the display screen 206. For example, the more button 212 may be configured to access advanced features such as menus, help and other applications. For example, the more button 212 may be configured such that, if it is pressed, the display screen 206 may present tabs for listing and launching applications installed on the first TCD, getting context-specific help, and listing functions provided by an application..

On every page of content that is presented on the display screen 206 as the user scrolls, one item may be highlighted. Such an item may be highlighted by a visual cue, for example, reverse video or other visual highlighting, or by an indicator positioned next to the item such as an arrow or icon.

Context-specific buttons 210 also may be provided. If an application is configured to use these buttons 210, then in accordance with the application, on each page of content displayed on the display screen 206 as the user scrolls, one or more items may be listed so that they align with the context-specific buttons 210. Pressing a context-specific button 210 corresponding to an item selects the corresponding item and allows the application to proceed. Optionally, the item corresponding to the pressed

context-specific button 20 is selected for further application processing even if another item is currently highlighted by a visual cue such as reverse video.

Thus, the function of each button 210 depends on the content being displayed on the display screen 206. Specifically, each button 210 corresponds to an item of content being displayed at a particular location on the display screen 206.

For example, referring to Fig. 6, the display screen 206 is displaying a list of applications. The buttons 210 along the left and right-hand side of the display screen 206 each correspond to a particular application item. If a user presses one of these buttons, the application corresponding to the button will be executed, which may result in the display screen 206 now displaying new content specific to that application.

The content-specific buttons 210 along the bottom of the display screen 206 may determine the body of content to be displayed on the display screen 206. For example, in Fig. 6, each button 210 corresponds to a tab at the bottom of the display screen 206, where each tab corresponds to a different body of content to be displayed.

A first application may configure the display screen 206 such that if the content, e.g., a list, to be displayed by a second application does not fit on the display screen, a slider may indicate on the display screen 206 a position of the content being displayed relative to the entire content to be displayed by the second application.

Figs. 5 and 6 illustrate example embodiments of a telephone and a user interface for implementing a TCD. These embodiments are for illustrative purposes, as several other embodiments of telephones having any of a variety of configurations may be used to implement a TCD as described herein.

3. Communications Network

The communications network on which the first TCD resides may have any of a variety of configurations. For example, the communications network may include merely a network medium and two or more TCDs. Although the communications network is primarily described herein as being a wire-based network having wire (i.e., cable) as the network transmission medium, the communications network may be any of a variety of types of networks adhering to any of a variety of wireless protocols, e.g., protocols for PCS networks, protocols for 3G networks or the wireless Ethernet protocol defined by IEEE 802.11, and having any of a variety of network transmission media, such as carrier waves and fiber optic cables. Further, as is described below in relation to Fig. 7, the communications network may include a plurality of sub-networks or

segments, where each sub-network may include any of a plurality of transmission mediums.

Fig. 7 is a block diagram illustrating an example embodiment of a communications network 300. The communications network 300 may include a network transmission medium of 332 and a plurality of TCDs including 302 and 306. The communications network 300 may be configured such that no other network resources are necessary to conduct a telephone call between two or more of the TCDs. For example, each TCD may be configured such that all of the telephony applications to be executed on the TCD reside completely on the TCD. Further, each TCD may be configured such that all of the data necessary to set-up a telephone call, including the network address of each participant of the telephone call, and any data necessary to execute any applications, also are stored entirely on the TCD.

Alternatively, the communications network 300 may be more distributed, including one or more other network resources to store data applications and parts of applications, and to assist in performing one or more telephony applications.

Accordingly, the communications network 300 may include one or more companion devices, for example, companion devices 304 and 310, one or more directory databases, including directory database 308 and directory database 320, a TCD deployment server 314, one or more application servers 318, one or more call control servers 322, one or more network interfaces, for example, an IP/PSTN Gateway 326, and one or more management databases, for example, management database 316.

The communications network 300 may be divided into a plurality of sub-networks or segments. For example, network elements 302, 304, 308 and 310 may reside on a network transmission medium 334 separated from network transmission medium 332 by one or more network interface elements 312. The one or more network interface elements 312 may be routers, bridges, switches, media gateways, microwave transmitters/receivers, cellular PCS network elements, or any combination thereof that control the transmission of data between network transmission mediums 334 and 332.

For example, network elements 302, 304, 306, 308, 310 and 334 may be part of a customer network 340, and network elements 314, 318, 322, 326, 316, 320, 323 and 332 may be part of a service provider network 324.

The communications network 300 may include an IP/PSTN Gateway 326 for setting up a phone call between a TCD on the communications network 300, e.g., TCD

302, and a TCD on the PSTN, for example, TCD 334. Depending on the configuration of communications network 300, a different type of gateway may be used to communicate with TCD 334 on the PSTN.

The IP/PSTN Gateway 326 converts media and call control messages conforming to one or more IP protocols and received from the communications network 300 to media and call control messages conforming to PSTN protocols to be sent onto the PSTN. Conversely, the IP/PSTN gateway may be configured to also convert PSTN media and call control messages from the PSTN into IP media and call control messages to be transmitted onto the communications network 300.

As described above, a TCD 302 may be any of a plurality of devices including a telephone, for example, the telephone 200 of Fig. 5, or a computer. If the first TCD is a telephone, it may desirable to provide a companion device to provide additional resources for the telephone, including additional telephony functionality, data storage, and an environment for developing telephony applications.

For example, at a user's home or office, a user may have a TCD embodied as a telephone for participating in telephone calls, and may have a general purpose computer for a variety of other tasks. Although the user's telephone may have more desirable ergonomic features for participating in the telephone call, the general purpose computer may have a plurality of features more desirable for developing applications and storing, retrieving, and manipulating data. For example, a general purpose computer may include peripherals such as a video monitor, a keyboard and a mouse that the telephone embodiment of the TCD lacks. Further, the general purpose computer may have more extensive memory for storing data and applications and for running applications.

To develop applications for a TCD, a companion device may include a plurality of APIs, including the APIs described above in relation to the open API layer 5 and the core telephony functionality API layer 21. To load these applications onto a TCD, and unload applications from the TCD, a companion device may include an applications manager such as that described above in relation to the system API 7.

A companion device also may be loaded with management software for managing various aspects of the communications network 300. For communicating and sharing data with other network resources, particularly TCDs, a companion device may be configured to communicate using a number of protocols, including the Simple Network Management Protocol (SNMP), the Hypertext Transport Protocol (HTTP),

Remote Method Invocation (RMI), the Hypertext Mark-up Language (HTML), and the File Transfer Protocol (FTP).

The directory database 308 may store a variety of information about other users, TCDs, other network resources, or the communications network 300 itself. Thus, a user may access this information to determine how to reach another user to set-up a telephone call. For example, the directory database 308 may store data for a commercial directory or phonebook product such as Microsoft Outlook® available from Microsoft Corporation or Lotus Notes® available from Lotus Development Corporation. These commercial software products may reside on the TCD itself or on a companion device to the TCD and may be configured to access the directory database 308 for information.

Other directory databases, such as the directory database 320, may store data for one or more directory servers 323, as is described below in more detail.

As described above in relation to Fig. 1, one or more telephony applications may be stored and executed entirely on a TCD, one or more applications may be distributed such that at least part of the application is stored on another network resource, and one or more applications may be stored remotely from the TCD on another network resource, where the TCD maintains pointers to the remote applications. The one or more application servers 318 may serve as network resources that store one or more applications and parts of at least one or more applications.

For example, an application server 318 may include a voicemail application or the server side of a client/server application, where the client side resides on one or more TCDs. Further, although a TCD may support conference bridging, as described above in relation to Figs. 3 and 4, if the number of connections involved in the conference call becomes too great, it may be more efficient to perform the conference bridging with a conference bridging application resident on an application server 318 that has more resources, or resources more dedicated to conference bridging.

The communications network 300 may include one or more directory servers 323 for directing calls between TCDs. Each directory server 323 may include a plurality of directories or indexes that map a user identifier such as a telephone number, logical name or extension name to a network address, for example, an IP address. This network address may be the network address of a TCD assigned to the user, or may be an address of another directory server that may store the network address of the TCD assigned to the user, or may be the network address of an IP/PSTN Gateway, for example, IP/PSTN

Gateway 326. One or more of these directories may be stored on the directory database 320 that is accessible by the directory server 323.

For example, a first network address directory may include a plurality of entries, each entry corresponding to a telephone number, for example, 123-456-7890. Each entry may store a network address, for example, 10.20.30.40, corresponding to the telephone number.

As a result of the first network address directory receiving a telephone number, the first network address directory may lookup and access the entry in the directory server corresponding to the telephone number and retrieve the IP address contained therein. A first network address directory, as well as the second and third network address directories described below, may be configured to use this retrieved IP address in any of a variety of ways, as described in more detail below.

The directory server 323 or the directory database 320 may also include a second network address directory, where each entry of the a second network address directory corresponds to a logical name, e.g., jsmith@acme.com, of a user of the communications network 300. Each entry may store an IP address associated with the logical name. Upon receiving a logical name, the directory server 323 may lookup and access the entry corresponding to the logical name and retrieve the IP address stored therein.

The directory server 323 or directory database 320 also may include a third network address directory, where each entry corresponds to an extension, for example, x1234@acme.com. Each entry may include an IP address associated with the extension. Upon receiving an extension, the directory server 323 may look up and access the entry corresponding to the extension and retrieve the IP address stored therein.

Although the network address directories have been described above as separate entities, any combination of these network address directories may be combined to form one or more directories.

As described above, a TCD may store or have direct access to one or more network addresses corresponding to one or more TCDs, respectively. If a first TCD stores or has direct access to the network address of a second TCD, the first TCD may set-up a call with the second TCD directly by using a peer-to-peer protocol such as SIP or H.323.

For example, TCD 302 may contact TCD 306 directly by sending a SIP set-up message to TCD 306, and TCD 306 may reply directly to TCD 302 to inform TCD 302

that it is willing to participate in the telephone call. Lastly, TCD 302 may send an acknowledgement message to TCD 306 to acknowledge that it received the response message from TCD 306. An analogous method may be applied to establish a telephone call between TCD 302 and TCD 334, except that the IP/PSTN Gateway 326 would serve as a network interface between communications network 300 and the PSTN.

Alternatively, TCD 302 may not store or have direct access to the network address of TCD 306. The TCD 302, however, may store or have direct access to other user identifiers of the TCD 306, for example, a telephone number, logical name or extension of TCD 306. Accordingly, the TCD 302 may communicate with a directory server 323 to determine the IP address corresponding to one or these indicators, as described above.

If TCD 302 sends the user identifier to the directory server 323, the directory server 323 retrieves the network address corresponding to the TCD 306, depending upon the configuration of the directory server 323 and the TCD 302, the directory server 323 may take any other plurality of actions. For example, if the directory server 323 is configured similar to a SIP proxy server, the directory server 323 may forward a call set-up message to TCD 306, which would respond to the directory server 323, which would then forward the response message to TCD 302.

Alternatively, if the directory server 323 is configured similar to a SIP redirect server, the directory server 323 may send the determined network address back to TCD 306, which would then send a call set-up message to the determined network address.

As discussed above, however, each entry of a directory server 323 also may include an IP address of another network resource e.g., another directory server. The directory server 323 may be configured to send the received identifier to the other network resource to perform further look-ups to determine the network address of the user. If the network address retrieved is the network address of another directory server, and the directory server 323 is configured similar to a SIP proxy server, the directory server 323 may forward the call set-up message to the other directory server, and repeat the lookup process. Such communications between directory servers, and the lookups performed thereon, may be repeated until the network address of the user is determined.

If the network address retrieved by the directory server 323 is the network address of another directory server, and the directory server 323 is configured similar to a SIP redirect server, the directory server 323 may send the retrieved network address to

the TCD 302. The TCD 302 then may access the other directory server, and repeat the lookup process. Such communications between directory servers and the TCD 302, and the corresponding lookup process, may be repeated until the network address of the user is determined.

Further, an entry in the directory server may include multiple network addresses corresponding to a single identifier. For example, for a customer service telephone number, the customer service telephone number may map to a plurality of network addresses. Consequently, the directory server 323 may include logic to determine which network address to contact, and either send a call set-up message to the determined address, or send the determined address to the TCD 302.

As described above, for telephone numbers corresponding to a TCD that reside on the PSTN, for example TCD 334, the directory server 323 may retrieve an IP address of an IP/PSTN Gateway, for example, the IP/PSTN Gateway 326. The network address retrieved by the directory server 323 may determine a network address of an IP/PSTN Gateway, for example, the network address of IP/PSTN Gateway 326 that is located most closely geographically to TCD 334. Depending on the configuration of the directory server 323, the directory server 323 may either forward a call set-up message to another network resource that will forward the message along to the appropriate IP/PSTN gateway, or may send the network address to the TCD 302 and allow the TCD 302 to contact the appropriate IP/PSTN Gateway 326 on its own.

If the TCD 302 is configured to use a master/slave call control protocol to set-up telephone calls, then, to set-up a telephone call, the TCD 302 may send a call set-up message to a master/slave controller 322. The master/slave control 322 then may set-up the telephone call using the directory server 323 and directory database 320 using analogous techniques to those techniques described above with respect to a TCD 302 configured to use a peer-to-peer protocol.

Similarly, for telephone calls coming into the communications network 300, for example, by IP/PSTN Gateway 326 or by a network switching element such as a switch, router, bridge or hub, the directory server 323 may be contacted to determine a network address of a call recipient. For example, if TCD 334 attempts to set-up a call with TCD 302, the IP/PSTN Gateway 326 will convert the set-up message to a set-up message conforming to a default call control protocol being used by the communications network

300. The IP/PSTN Gateway 326 will then forward the call set-up message including the user identifier to the directory server 323 to determine the network address of TCD 302.

Several other techniques may be used to direct telephone calls to the appropriate user and TCD.

A TCD that the TCD 302 is trying to set-up a call with may be configured to set-up calls using a call control protocol different than the call control protocol of the TCD 302. For example, the TCD 302 may be configured to use the SIP protocol and TCD 306 may be configured to use the H.323 protocol. Accordingly, the directory server 323 may be configured to determine the call control protocol that a network address supports and send this information to the TCD 302, or the TCD 302 may be configured to determine that a network address returned by the directory server 323 corresponds to a certain call control protocol.

If the TCD 302 is configured to be able to communicate using the H.323 protocol, then TCD 302 may communicate with TCD 306 using the H.323 protocol. If, however, TCD 302 is not configured to set-up a telephone call using H.323, then TCD 302 may send the SIP message to an H.323/SIP signaling gateway. The H.323/SIP signaling gateway may convert the SIP message to an H.323 message. Accordingly, the H.323/SIP signaling gateway can send the generated H.323 message to TCD 306, and when TCD 306 responds with an H.323 message, the H.323/SIP signaling gateway may convert this H.323 message to a SIP message and send the SIP message to TCD 302.

In an embodiment of the communications network 300, the network 300 may be configured to identify a call agent for each call that comes in to the network. For inbound calls, the associated call agent may be a SIP proxy server, an MGCP call agent, an H.323 gatekeeper, or an agent used by another call signaling protocol.

As a number of TCDs 302 on the communications network 300 grows, it may be desirable to delegate telephony applications that may be provided by TCDs themselves to other network resources, for example, TCD deployment server 314. The TCD deployment server 312 may be configured with a plurality of telephony applications that also may be configured on a TCD. The TCD deployment server 314 may include applications to configure and manage TCDs, perform software upgrades on TCDs, determine which applications should be installed on which TCDs, configure a plurality of TCDs into hierarchical groups that may have multiple tiers of hierarchy, for example: geographical region, metropolitan region, business division, business department,

building, floor, and group, down to a user. The TCD deployment server also may provide several other services to a TCD.

The TCD deployment server 314 also may maintain personal TCD configuration information specific to a user of the communications network 300. Accordingly, regardless of where on the network a user uses a TCD, the user may indicate a unique identifier to the TCD deployment server 314, and the TCD deployment server 314 may configure the TCD that the user is using according to the personal configuration information of the user.

The TCD deployment server 314 may store the applications and data described above in a management database 316. The management database 316 may be any of a plurality of types of databases including an object-oriented database, a relational database, or a flat file database. Further, the TCD deployment server 314 may be configured to access the management database 316 using a plurality of techniques, including Open DataBase Connectivity (ODBC) techniques or Java DataBase Connectivity (JDBC) techniques. Further, to communicate with other network resources, including one or more TCDs and companion devices, the TCD deployment server 314 may be configured to communicate using a number of protocols, including SNMP, HTTP, RMI, HTML, and FTP.

The communications network 300 may be configured in any of a variety of ways, and the configuration of Fig. 7 is provided for illustrative purposes only. Several other network configurations may be used.

4. Telephony Applications

As described above, the open and extensible telephony system architecture 1 provides the ability to develop applications using the core telephony functionality layer 9 and the open API layer 5, respectively. Accordingly, the plurality of applications now described may be implemented using the open API layer 5, the core telephony functionality layer 9 or any combination thereof. Further, such applications may be run on one or more TCDs, one or more other network resources, or a combination thereof.

Several of the applications described below define sending/receiving information, including media, e.g., audio or video, textual data, instructions, or other information to/from one or more TCDs either during a telephone call or to set-up a telephone call. The information that is sent/received may depend on a state of the first TCD or the other

TCDs. Further, the action taken by a TCD as a result of receiving this information may depend on a state of the receiving TCD.

Further, each of the applications defined below may be configured such that the application may be added to the first TCD or modified after the first TCD has been deployed on the communications network, for example, by using one or more APIs of the open API layer 5, as described above.

Further each of the applications described below may be embodied as computer signals on any of a variety of computer-readable mediums, for example, a readable and writeable nonvolatile recording medium, such as a disk, a flash memory, a memory stick, a PC Card or a tape. Such a disk may be removable, for example, a floppy disk or a read/write CD, or permanent, known as a hard drive.

Accordingly, the applications described below may access one or more abstractions of the core telephony functionality layer 9 to determine a state of a TCD or a telephone call on the TCD, and may control one or more media processing elements of the media processing component 30, described above, to process media in accordance with the application.

A first telephony application may be configured to permit a caller from a TCD to have control over the sound that plays on the callee's TCD to indicate an incoming call from the caller. The sound may be played from an audio file accessible by the callee, for example, an audio file stored on audio storage medium 60, may be coded within the tone generator 69 of the callee's TCD, or the sound itself may be sent from the caller's TCD to the callee's TCD. The sound may be a ring, a voice announcement, or any sound that will be meaningful to the person being called. The sound may play intermittently until canceled by the callee, and may be broadcast over multiple TCDs on the communications network, for example multiple customer service TCDs. Such an application also may be used by the caller to "page" another TCD, but not set-up an actual telephone call.

The sound to be played may be pre-recorded, for example, by capturing the sound through the phone mouthpiece or through a direct electrical audio signal capture jack on the phone, and then storing the sound on the TCD itself or on an accessible network resource. Both the callee's and caller's TCD may be configured to control the manner by which the sound is played, for example, by a programmer of the telephony application or by the caller or callee through user parameters provided by the telephony application.

Fig. 8 is a flowchart illustrating an example embodiment of a method of indicating an incoming call to at least a second user of a second TCD. Such a method may be defined by one or more telephony applications.

In Act 452, a dialing instruction corresponding to a network address of the second TCD is received. Next, in Act 454, the first user is identified from a dialing instruction, and in Act 456, a sound to be played is selected based on the identification.

In a following Act 458, a call set-up message is sent to the second TCD at the network address. The call set-up message includes information to cause the second TCD to produce the selected sound. The information included in the call set-up message may include information representing the selected sound or instructions to play the selected sound. Further, the information may include a location from which to play the selected sound, for example, a URL.

Another telephony application may enable multiple audio inputs received during a conference call between TCDs to be selectively mixed or muted, as described above in relation to the local call bridge 51 of Fig. 3. Accordingly, a user of the first TCD participating in a conference call may selectively mute or enable the audio streams of other conference call participants. Further, such an application may be configured to permit a user on a second TCD that is not controlling the conference call to send a communication to the first TCD instructing the first TCD to selectively mix audio received from and/or sent to the second TCD. As a result, announcements can be made to a selected subset of participants and questions directed to another subset, or the comments of specific individuals screened out of the conference temporarily.

Although selectively mixing and muting audio signals has been described herein primarily in connection to bridged conference calls, similar techniques may be used to selectively mix and mute audio for fully-meshed conference calls, where each TCD maintains a representation of every connection included in the conference call. Although such a fully-meshed conferencing configuration reduces the load on a single TCD (i.e., the TCD that would otherwise provide the bridging), this configuration creates more network connections thus increasing network traffic and also makes controlling the conference call more complicated as now multiple TCDs are involved.

Further, to implement selective muting and mixing during a telephone call, one or more specialized versions of a TCD may be provided, each containing extra signal processing power for mixing/muting.

Fig. 9 is a flowchart illustrating an example embodiment of a method 460 of selectively transmitting media during a conference call including a first user input for receiving media from the first user and two or more connections between the first TCD and two or more second TCDs, respectively. Each second TCD corresponds to a second user. Such a method may be defined by one or more telephony applications.

In Act 462, a user selection is received that identifies one or more third users to include in receiving media from one or more fourth users. Each third user and fourth user is either the first user or one of the second users.

In a next Act 464, during the conference call, one or more media inputs are received from one or more of the fourth users. In a following Act 466, during the conference call, the one or more third users are permitted to receive media produced from the one or more media inputs.

Fig. 10 is a flowchart illustrating another example embodiment of a method 470 of selectively transmitting media during a conference call that includes a first user input for receiving media from the first user and two or more connections between the first TCD and two or more second TCDs, respectively. Each second TCD corresponds to a second user. Such a method may be defined by one or more telephony applications.

In Act 472, a user selection is received that identifies one or more third users to exclude from receiving media from one or more fourth users. Each third and fourth user is either the first user or one of the second users.

In a next Act 474, during the conference call, one or more media inputs are received from one or more of the fourth users. In a following Act 476, during the conference call, the one or more third users are prevented from receiving any media produced from the one or more media inputs.

In both methods 460 and 470, the media produced from the one or more media inputs may be media mixed in any of the variety of ways described above. Further, for both methods 460 and 470, the one or more third users may be permitted or prevented from, respectively, receiving media by using the local call bridge 51 of Fig. 3 as described above. Also, for both methods 460 and 470, the received user selection may be received from the first user of the first TCD or from one of the second users.

Another telephony application may define a method of communicating, as part of a telephone call, textual information from a first user of the first TCD to a second user of a second TCD connected to the first TCD by a first connection. Specifically, during the

telephone call, textual data defining text may be received on the first connection from the second TCD. The first TCD may then display the text defined by the textual data, for example, on display screen 206 of Fig. 5 described above. Such textual data may be any of a variety of data, for example, information about the second TCD or a user of the second TCD.

A scheduling application may be defined to allow a user to schedule a telephone call, for example, a conference call. Using a user interface as described above in relation to Figs. 5 and 6, or another network resource such as a companion device 304 of Fig. 7 described above, a user can schedule a telephone call by inputting telephone numbers of each of the telephone call participants and the date and time of the call. Further, if a remote conference bridge, i.e., not the local call bridge 51, is be used to perform the telephone call, the user may configure the scheduling application to call the remote conference bridge to set-up the conference call.

Such a scheduling application may interface with and access data from a database such as that provided by Microsoft Outlook® or Lotus Notes®, and may be configured to use the Internet Calendar and Scheduling Protocol (ICalendar) for communicating between the first TCD and any other TCDs or network resources involved in scheduling the conference call.

Such a scheduling application may be configured such that after all the telephone call information, including the identifications of the participants and the date and time of the call, has been entered, the application automatically notifies the other participants about the scheduled call. This notification may provide a variety of information to the other participants, including the date and time of the call, and the identities and information regarding the other call participants.

The scheduling application may be configured to notify the other participants in the telephone call by sending an electronic mail (email) message, for example, an e-mail message conforming to the Simple Mail Transfer Protocol (SMTP).

Fig. 11 is a flowchart illustrating an example embodiment of a method 480 of scheduling and performing a telephone call between a first user of the first TCD and one or more second users of second TCDs. In Act 482, one or more user identifiers are received, where each user identifier identifies one of the second users. Such a method may be defined by one or more telephony applications.

In a next Act 484, date and time information is received that specifies a future date and time to conduct the call, and in Act 486, the user identifiers and date and time information may be stored. The identifiers and date and time information may be stored locally on the first TCD or on a remote network resource accessible by the first TCD.

In a following Act 488, at the date and time specified by the date and time information, the user identifiers are accessed. Next, in Act 490, a telephone call may be initiated between the first TCD and the one or more second TCDs.

Method 480 also may include acts of receiving a conference bridge identifier identifying a conference bridge, wherein the act of initiating the conference call includes calling the conference bridge. Further, the method 480 may include notifying the one or more second users about the call before initiating the telephone call by sending a message including information about the call from the first TCD to the one or more second TCDs.

A web-based dialing application may provide web-based dialing by defining a web server embedded within the first TCD to set-up telephone calls. The web-based dialing application may be configured to allow a user to create a web page on the embedded web server that includes links to the URLs that represent one or more participants to include in a telephone call. A link may include a variety of information about a telephone call, including an identifier, e.g., a telephone number, for each participant and an identifier of a conference bridge.

Using a web browser application on the PC or other network resource, a user may access the web page on the first TCD and click on one or more URL links to cause the TCD to dial the selected number(s). The web-based dialing application may be configured to enable a user to dial one or more TCDs on different types of networks, including telephones on the PSTN and/or on a data network. Further, the web-based dialing application may define a call set-up HTML tag, e.g., "callto:<destination>", that operates similarly to the mailto:<destination> HTML tag, such that clicking the call set-up tag triggers the creation of a telephone call to the <destination>.

An example web-based dialing application may define a method of initiating a telephone call between a first user of the first TCD and one or more second users of second TCDs. This method may include receiving one or more URL selections made by the first user, where each URL selection corresponds to one of the second TCDs. Next, a telephone call between the first TCD and the one or more second TCDs may be initiated.

Such a web-based dialing application may use one or more network directories, as described above in relation to Fig. 7, to map a user of TCD identifier, e.g., URL, telephone number, extension, to determine a network address for which to set-up a telephone call. Further, such a web-based dialing application may use other network resources, e.g., an IP/PSTN gateway, to send a set-up message to a TCD located on another network, e.g., the PSTN, as described above in relation to Fig. 7.

Another telephony application may be defined to display text on a display screen of a first TCD, where the content of the text corresponds to the content of an audio signal being received on the first TCD. For example, such a telephony application may be configured to display options on a display screen, e.g., display screen 206 corresponding to options played by an Interactive Voice Response (IVR) system. For example, such a telephony application may be configured to indicate to a second TCD (e.g., a TCD that is part of an IVR system) that the first TCD is capable of displaying text. In response, the second TCD may send, or control another resource to send, text content corresponding to the audio content to the first TCD.

The first TCD then may display the text content, for example, a list of choices, on the display screen, as the IVR message is being played. A user may then make a selection using a user interface corresponding to the display screen, e.g., the user interface described above in relation to Fig. 5, as opposed to waiting for the complete IVR message to be played on the first TCD.

If the second TCD is part of an IVR system, e.g., a customer service network, a central server may act as a proxy for the second TCD and other TCDs of the IVR system and may be the network resource that sends the text signal to the first TCD.

Fig. 12 is a flow chart illustrating an example method 500 of displaying text associated with media, e.g., audio or video, received on a first connection between the first TCD and a second TCD as part of a telephone call, where the first TCD includes a display screen, for example, display screen 206 of Fig. 5. Such a method may be defined by one or more telephony applications.

In Act 502, a media signal may be received on the first connection from the second TCD, where the media signal represents media content.

In Act 504, before or concurrently to receiving the media signal, a textual signal is received on the first connection. The textual signal represents text content corresponding to the media content.

Next, in Act 506, the text content is displayed on the display screen. In a following Act 508, after or concurrently to displaying the text content, the media content may be played.

Other applications may be defined to send textual data in response to a call set-up message. For example, such an application may be configured such that upon receiving a call set-up message from another TCD, the first TCD responds with a textual message, where the textual message depends on a state of the first TCD. For example, the first TCD may be defined to have a "not in office" state. Accordingly, the application may be configured such that when the first TCD receives a call set-up message, the first TCD sends a list of other contact methods to the other TCD, for example, a cell phone number, a voice mail address, or a telephone number of another user.

Fig. 13 is a flow chart illustrating an example embodiment of a method 510 of communicating information in accordance with a state of the first TCD to a second TCD.

In Act 512, a call set-up message is sent to the second TCD to set-up the telephone call. Next, in Act 514, in response to the call set-up message, a textual signal is received from the second telephony communication device. The textual signal defines text content corresponding to a state of the second TCD.

In a following step 516, the text content is displayed on the display screen of the first TCD.

The text content may include one or more options corresponding to the state of second TCD, where each option indicates an action to be taken in response to the state of the second TCD. These one or more options may be displayed to the user of the first TCD. Accordingly, the method 510 also may include an act of receiving a selection of one of the options from the first user, and sending a selection signal representing this selection to the second TCD to perform the action indicated by the selection.

Fig. 14 is a flow chart illustrating an example embodiment of a method 520 of communicating information in accordance with a state of the first TCD to a second TCD.

In Act 522, a call set-up message to initiate a telephone call is received at the first TCD from the second TCD. Next, in Act 524, a state of the first TCD is determined.

In a following step 526, the first TCD transmits a textual signal representing the determined state to the second TCD for display.

As described above, the textual signal may include textual representations of one or more options for a user of the second TCD to be displayed on the second TCD. Each

option may indicate an action to be taken in response to the state of the first TCD. Accordingly, the method 520 may further include an act of receiving a selection signal from the second TCD, where the selection signal represents a selection of the one or more options by the user of the second TCD, and an act of performing the action indicated by the selection.

Another application may define a method of communicating an audible expression during a telephone call on the communications network, where an audible expression is a sound that has meaning to a user. Such an audible expression may express an emotional state or provide emphasis during a phone call. For example, an audible may be a sound of a bomb exploding, a popular expression, a bell gonging, or another sound that would have meaning to a user. These sounds may be encoded as part of the tone generator 69 or may be stored on a storage medium accessible by the first TCD, e.g., audio, storage medium 60. The application may be configured to mix the audible expression and a user's voice on the first TCD such that another participant in a phone call will hear a mixed audible signal including the voice and the audible expression. Optionally, the audible expression can be mixed with a user's voice such that only the audible expression is heard.

Video or textual expressions also may be used in a similar manner to communicate during a telephone call.

One or more applications may be defined to play and stop media on the first TCD as part of a game. For example, a "whack-a-mole" style game may be defined such that a media, e.g., audio or video, plays on the first TCD until a user of the first TCD selects a particular key sequence or presses a particular button on the user interface of the first TCD. If a correct sequence is entered by the user within a set period of time, then the playing of the sound may be stopped and control of the game may be sent to another TCD to continue the game.

Further, an application may be configured such that a quote-of-the-day or fortune is received on the TCD at specified intervals, e.g., daily, weekly, hourly, as configured by a user or a network provider.

Other telephony applications may be defined to play text, media, or other information on the first TCD as part of an advertisement or as part of a screen saver that runs on a display screen of the first TCD.

Fig. 15 is a flow chart illustrating an example embodiment of a method 530 of communicating an audible expression during a telephone call.

In Act 532, a voice signal representing a voice of a first user of the first TCD is received. In a next Act 534, a selection signal representing a selection of an audible expression by the first user is received.

Next, in Act 536, the audible expression signal corresponding to the selection signal is generated, where the audible expression signal represents the audible expression. In a following Act 538, the audible expression signal and the voice signal are mixed to produce a mixed signal.

Next, in Act 540, the mixed signal is transmitted as part of the telephone call to one or more second users at one or more second TCDs.

A call screening application may be defined to screen incoming calls to the first TCD based on data associated with the incoming call, for example, a caller identification (i.e., caller ID), the time of day, or other associated data. The screening application may define rules, or provide parameters by which a user may define rules, to determine how to respond to a call set-up message based on different criteria. For example, for a highly important call, the rules may be defined such that the first TCD tries a series of different phone numbers in an effort to contact a user of the first TCD. Further, the screening application may be defined such that, for less important calls, the first TCD may offer voicemail or other alternatives. Also, the screening application may be configured such that the caller may be queried to supply additional information before the call is filtered appropriately.

Fig. 16 is a flow chart illustrating an example embodiment of a method 550 of screening a telephone call based on information associated with the call.

In Act 552, a first call set-up message is received at the first TCD from a second TCD. Next, in Act 554, one or more call answering rules are accessed, where each call answering rule defines a condition and an action to be taken by the first TCD in response to receiving a call set-up message if the condition is satisfied.

In a following Act 556, for one or more of the call answering rules, it is determined whether the condition is satisfied. Next, in Act 558, the call set-up message is responded to in accordance with the determinations for the one or more call answering rules.

As described above, a first rule of the one or more rules may define an action to be taken on condition of a call set-up message being received during one or more time intervals. Accordingly, the method 550 also may include acts of determining a first time at which the call set-up message is received, and comparing the first time to the one or more time intervals to determine whether the first time is within one or more of the time intervals. If the first time is within one of the one or more time intervals, the act of responding to the call set-up message comprises responding to the call set-up message in accordance with the action defined by first the rule.

Further, as described above, a first rule of the one or more rules may define an action to be taken on condition of the call set-up message being from a particular user. Accordingly, the act of receiving of method 550 may include receiving an identification signal corresponding to the call set-up message, where the identification signal identifies a user of the second communication device. Further, the act of determining may include identifying the user from the identification signal, and comparing the identification of the user to identifications of one or more particular users to determine if the user is one of the particular users. If the user is one of the particular users, the act of responding to the call set-up message may comprise responding to the call set-up message in accordance with the action defined by the first rule.

Further, one or more of the call answering rules may define forwarding the call set-up message to one or more other TCDs if a first condition is satisfied. Accordingly, the act of responding may include forwarding the call set-up message to the one or more other TCDs if the first condition is satisfied.

Further, the screening application may be configured such that one of the call answering rules define playing a sound if a first condition is satisfied. Accordingly, the act of responding may include playing such sound if the first condition is satisfied.

The screening application also may be configured such that one of the call answering rules defines prompting a user of a calling TCD for more information if a first condition is satisfied. Accordingly, the act of responding may include prompting the user for more information if the first condition is satisfied.

Another application may be configured to integrate e-mail with a telephone call. Fig. 17 is a flow chart illustrating an example embodiment of a method 560 of playing content of an e-mail message as audio on the first TCD.

In Act 562, the electronic e-mail message is controlled to be sent to a network resource that converts the textual content of the e-mail message to a voice signal. Optionally, the first TCD may include a text-to-speech application, such that the first TCD is the network resource.

In the next Act 564, a call is set-up between the network resource and the first TCD. In a following step 566, the voice signal is controlled to be transmitted to the first TCD as part of the telephone call. This transmission may be controlled by the first TCD.

Next, in Act 568, the voice signal is played on the first TCD.

Another telephony application may be configured to record a telephone call conversation and then send the recorded conversation to one or more audio-playing devices, e.g., one or more other TCDs. For example, such a telephony application may define a method of communicating, to one or more second users of one or more second TCDs, at least a first part of a telephone call between a first user of the first TCD and one or more third users of one or more third TCDs. Such a method may comprise an act of recording as an audio file at least the first part of the telephone call between the first user and the one or more third users, and sending the audio file to the one or more second TCDs. The audio file may be sent by attaching it to an e-mail message and sending the e-mail message to the one or more second TCDs, or by attaching the audio file to a call set-up message and sending the call set-up message to the one or more second TCDs.

Another telephony application may define a method of sending a graphical file as a message for a user of a TCD. Fig. 18 is a flow chart illustrating an example method 570 of sending a graphical message to a first user of a second TCD.

In Act 572, a call set-up message is sent from the first TCD to the second TCD. Next, in Act 574, an indication is received from the second TCD that the first user is not answering the call set-up message.

In a following Act 576, a graphical file is sent from the first TCD to the second TCD to store as a message to the first user. The first user then may display the message on a display screen on the second TCD. Optionally, the second TCD may be configured such that only specific users can leave graphical messages on the second TCD.

Fig. 19 is a flow chart illustrating another example embodiment of a method 580 of communicating a graphical message to a user of a first TCD, where the first TCD includes a display screen.

In act 582, a call set-up message is received from a second TCD. In a following step 584, an indication that the first user is not answering the call set-up message is sent.

Next, in act 586, in response to the sending of the indication, a graphical file is received from the second TCD as a message.

Next, in act 588, the graphical file is displayed on the display screen to communicate the message to the first user.

Method 580 also may include an act of storing the graphical file on a storage medium accessible by the first TCD, for example, a local storage medium on the first TCD or another storage medium located on another network resource.

Another telephony application may be configured to add a connection to a media source, for example, a media source on the Internet, and feed the media from that media source to another TCD that is put on hold by the first TCD. Such media may be audio, video, or a combination thereof.

Fig. 20 is a flow chart illustrating an example embodiment of a method 590 of placing one or more connections on hold during a telephone call.

In Act 592, an instruction to place a telephone call on hold is received, and in Act 594, an indication of a location of a media file is received.

In Act 596, a connection to the location of the media file is created, and in Act 598, media signals from the media file are received.

In a following step 600, the media signals are sent to the one or more connections on hold.

Another telephony application may be defined such that personal information about a first user, including configuration information about how to configure a TCD for the first user, and information about usage habits of the first user.

Accordingly, when a first user uses any of a plurality of TCDs on the communication network, the first user may provide the first users identification and the personal information of the first user may be accessed and used to configure the TCD that the first user is using. Optionally, this personal information may be stored on a network resource accessible by any of the plurality of TCDs on the communications network. Further, a network resource, for example, the TCD deployment server 314 described above in relation to Fig. 7, may be configured to retrieve the personal information for any of the TCDs on the communications network.

Fig. 21 is a flowchart illustrating an example embodiment of a method 610 of dynamically configuring a telephony communication device for a first user.

In Act 612, a user identification identifying the first user may be received, and an Act 614, configuration information specific to the first user may be accessed.

In a following Act 616, the TCD may be configured in accordance with the configuration information.

Further, this personal information may be made available to callers whenever a call is made, or may be saved into and accessed from a personal information management application such as Microsoft Outlook® or Lotus Notes®.

Also, select personal information may also be made available to a telephone call receiving IVR prompting. Accordingly, a telephone application may be configured such that, at each IVR prompt, the select personal information may be accessed, and a response to the IVR prompt automatically sent, or, if a security feature is enabled, sent after permission is granted.

Another telephony application may be configured to enable a TCD to distort, scramble or encrypt a media signal before sending the signal onto the communications networks.

Another telephony application may be configured to control the routing of a telephone call through specific service providers on the communications network based on pre-defined criteria, for example, rates. For example, if a call is to be set-up with a TCD from another network, such an application may be configured to query one or more IP/PSTN gateways or IP network-to-IP network service providers for rate information for such a call. Such an application may include logic to determine the IP/PSTN gateway or service provider through which to set-up the telephone call based on the rate information.

Another telephony application may be configured to provide hands-free call answering on a TCD. Such an application may be configured such that, when a call set-up message is received, an audible (e.g., ring) or visual signal (e.g., blinking light) indicates to the user that a call is incoming, the call is accepted, (i.e., a call is created on the TCD), and a speaker phone or other speaker is activated so that a user of the TCD instantly can begin a telephone conversation.

Another telephony application may be configured to apply voice stress analysis to a telephone call. Such an application may monitor the emotional state of a speaker using

known techniques and assign an associated numerical stress level. The numerical stress level then may be displayed on a display screen to a user of the first TCD participating in the telephone call.

Another telephony application may provide voice-based dialing on the first TCD. Such an application may include voice recognition logic to search a database of voice patterns corresponding to users' network addresses or other identifiers to search for a match to a name spoken by a user of the first TCD. Such a database may be located locally on the first TCD or remotely on another network resource. After a pattern match is made, the first TCD then automatically may set-up a call with the TCD corresponding to the match. Such an application may provide a user of the first TCD with access to any telephone number available on the Internet or any other telephone directory accessible by the first TCD merely by speaking a name into the first TCD.

As described above, a first TCD may store either locally or remotely telephone extensions and telephone numbers. Accordingly, a telephony application may be configured to search these numbers as a user is entering digits to call another user, and provide a list of numbers and extensions, or the names of users corresponding to the extensions and telephone numbers, on a display screen of the first TCD that match the digits entered so far by the user. The user then may select a telephone number extension or name from the display screen using a user interface, as opposed to entering the entire number into the first TCD. As each new digit is entered by a user, the list of matching telephone numbers may be reduced, and this process may be repeated until the user makes a selection from the list or the extension or telephone number is completely entered.

As described above in relation to Figs. 5 and 6, another application may provide an on-demand help system on the first TCD. Such a help system may provide step-by-step information to a user of the first TCD about how to use the TCD and how to use particular applications. Such a help application may be configured to use media files to help a user. For example, if a user is dialing an international telephone number, the help system may prompt the user for the name of the country that is being dialed. In response to the user's entry for the prompt, the help application may supply the associated country code and the expected digit length of the phone number. In another example, such a help application may provide a sequence of interactive text messages and audio clips that prompt a user through a new or infrequently used task.

A common problem in dialing a telephone number is incorrectly entering a digit, and having to hang up and redial the entire number. Accordingly, a telephony application may be configured to store all digits entered by a user locally before sending a set-up message to initiate a telephone call. Further, such an application may display the digits on a display screen as the user is dialing. Thus, such an application may enable a user who enters an incorrect digit to merely clear the digit from the screen and re-enter the correct digit before the telephone call is initiated. After the correct number is entered, the user may be enabled to hit a button or touch a location on the display screen to initiate the telephone call.

Another telephony application may be configured such that information about a user or TCD may be sent (e.g., attached) along with communication messages as part of a telephone call. Accordingly, such an application or a related application may be configured to extract such information from a communication message as part of a telephone call. Such information may include a local time at the caller's location or more sophisticated information such as a pointer to a web-based mapping application that visually shows the caller's location. Further, data representing the map itself may be sent along with a communication message. A variety of other forms of information may be sent along with a communication message as part of a telephone call.

Another telephony application may be configured to control a TCD to repetitively send a call set-up message to a network address at set intervals until the telephone call is answered by a human voice. Such an application may be used if a telephone call is too urgent to merely leave a voice message. Further, such an application may configure the first TCD to terminate a telephone call if the recipient at the network address transfers the telephone call in response to the call set-up message.

A space-saving application may define a method for loading only certain applications, portions of applications and data onto the first TCD in response to the TCD being initially deployed on a communications network. By configuring the first TCD to dynamically load only certain applications, portions of applications and data such a space-saving application limits the amount of memory consumed by applications on the first TCD, thereby conserving memory resources. This conserved memory then may be used for other purposes.

Another application may be defined to provide a method of resolving conflicts between a plurality of applications on the first telephony communication device. For

example, two or more applications may be defined to be notified of a same event. Accordingly, if the event occurs, a conflict occurs as to which application should be notified of the event.

Accordingly, such an application may be defined to receive an indication that a first application and one or more second applications are defined to be notified if a first event occurs, and to assign notification priority to the first application such that, if the first event does occur, the first application is notified of the event and the one or more second applications are not notified of the event. In other words, as described above in relation to the application arbitrator, the first application consumes the notification to prevent future conflicts with the one or more second applications.

This conflict resolution application may include a user interface to allow a user to make the selection and assign notification priority to an application. For example, the user interface may be implemented on a desktop PC or on a telephone such as the telephone described in relation to Figs. 5 and 6.

Another application may be defined, for a telephone call including a first user of a first telephony communication device and a plurality of second users of other telephony communication devices, a method of visually indicating to the first user that the voices of one or more third users of the plurality of second users are currently being played on the first telephony communication device.

The method may include displaying a plurality of user identifiers, each user identifier indicating one of the second users. For example, these identifiers may be listed on a computer screen or on a display screen of a telephone, e.g., the display screen 206 described above in relation to Figs. 5 and 6.

Audio data may be received from one or more of the second users on connections corresponding to the one or more second users, respectively.

For each of the one or more third users, voice data may be detected in the audio data received from the third user using known DSP techniques. The application may configure one or more of the media processing elements described above in relation to the media processing module 29 to perform the known DSP techniques to detect if a voice is received on a connection.

Next, the voice data of the third users may be mixed, for example, by the local bridge 51 of Fig. 3, to produce mixed audio data. In a following act, the mixed audio

data may be played on the first telephony communication device, for example, on a handset speaker, a headset speaker or base speaker.

To indicate to the first user that the voices of the one or more third users are currently being heard by the first user, for each of the one or more third users, a visual indicator may be displayed next to the identifier identifying the third user. For example, an icon may appear on the display screen next to the names of each second user for whom voice data was detected on the second user's connection. Other methods of indication may be used.

The performance of the DSP, including receiving the audio data, detecting the voice data, and mixing the audio data may performed on a different device than the device that displays the indicators described above. For example, a first TCD may perform the DSP and a companion device may display the indicators, or a remote server may perform the DSP, and the first TCD may display the indicators.

Further, a first device performing the DSP may communicate the voice detection, for example, as part of an RTCP message, to a second device that displays the indicators. Alternatively, the DSP and displaying may occur on a same device, for example, on the first TCD.

Having now described some illustrative embodiments, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by way of example only. Numerous modifications and other illustrative embodiments are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention. In particular, although many of the examples presented herein involve specific combinations of method acts or apparatus elements, it should be understood that those acts and those elements may be combined in other ways to accomplish the same objectives. Acts, elements and features discussed only in connection with one embodiment are not intended to be excluded from a similar role in other embodiments.

CLAIMS

1. A first telephony communication device that is part of a communications network that comprises a transmission medium, the first telephony communication device comprising:

a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium; and

a telephony software component to control operation of the hardware component, the software component comprising at least a portion of a telephony application defining a telephony function to be performed in association with a telephone call,

wherein the telephony application is modifiable after the first telephony communication device is deployed on the communications network with modifications developed independently from creation of the telephony software component.

2. The first telephony communication device of claim 1, wherein the telephony software component further comprises an open application programming interface to modify the telephony application.

3 The first telephony communication device of claim 1, wherein the telephony software component further comprises:

at least a part of a non-telephony application defining a telephony function to be performed independently of a telephone call.

4. The first telephony communication device of claim 1, wherein at least a part of an additional telephony application can be added to the telephony software component after the first telephony communication device is deployed on the communications network.

5. The first telephony communication device of claim 4, wherein the additional telephony application was developed independently from the vendor.

6 The first telephony communication device of claim 1, wherein the telephony application is written in a general purpose programming language.

7 The first telephony communication device of claim 1, wherein the general purpose programming language is Java.

8 The first telephony communication device of claim 1, wherein the telephony software component further comprises:

an operating system; and

an operating system interface to interface communications between the operating system and the telephony applications of the telephony software component such that the telephony applications are independent of the operating system.

9. The first telephony communication device of claim 1, wherein the telephony software component further comprises:

a call processing module to represent and control concurrently one or more telephone calls, each telephone call including one or more connections to other users corresponding to other telephony communication devices on the communications network, wherein, for each telephone call, the call processing module represents and controls each connection.

10. The first telephony communication device of claim 9, wherein, for each connection, the call processing module is operative to control communications on the telephony device corresponding to the connection using a first call control protocol selectable from a plurality of call control protocols available on the first telephony communications device.

11. The first telephony communication device of claim 10, wherein the telephony software component further comprises:

an application programming interface to develop applications for the first telephony communication device, wherein the application programming interface is generic to the plurality of call control protocols.

12. The first telephony communication device of claim 9, wherein the telephony software component further comprises:

a media processing module, the media processing module comprising, for each telephone call, a corresponding media processing component to represent media processing of the telephone call, and for each connection of the telephone call, the media processing component includes a corresponding media processing connection,

wherein, for each telephone call, to control the media processing of the telephone call, the call processing module controls the media processing component corresponding to the telephone call, and, for each connection of the telephone call, to control the media processing of the connection, the call processing module controls the corresponding media processing connection.

13. The first telephony communication device of claim 9, wherein the call processing module is operative to communicate with one or more other network resources on the communications network to control the one or more telephone calls.

14. The first telephony communication device of claim 13, wherein a second telephony communication device is connected to the communications network through a Public Switched Telephone Network and an IP/PSTN gateway, and

wherein the call processing module is operative to communicate with the IP/PSTN gateway to control, during the telephone call, communications between the first user and a second user of the second telephony communication device.

15. The first telephony communication device of claim 1, wherein the telephony software component further comprises:

a media processing module, the media processing module comprising media processing elements to define media processing functions;

wherein, to implement a media processing function, at least a first of the one or more telephony applications is operative to dynamically configure one or more combinations of the media processing elements during execution of the first telephony application.

16. The first telephony communication device of claim 1, wherein the telephony software component further comprises a pointer to a location on the communications network of an application remotely executable by the first telephony communication device.

17. The first telephony communication device of claim 1, wherein the telephony application defines a remote user interface for another network resource on the communications network.

18. The first telephony communication device of claim 1, wherein the first telephony communication device is a telephone.

19. A method of defining functionality for a telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium, and comprising a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call, the method comprising acts of:

after the telephony communication device is deployed on the communications network, accessing the telephony application on the telephony communication device; and

modifying the telephony applications with modifications developed independently from creation of the telephony software component.

20. A system for defining functionality for a telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium, and comprising a telephony software component to control

operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call, the system comprising:

means for accessing the telephony application on the telephony communication device after the telephony communication device is deployed on the communications network; and

means for modifying the telephony application with modifications developed independently from creation of the telephony software component.

21. A computer program product, comprising:

a computer-readable medium; and

computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a computer, instruct the computer to perform a method of defining functionality for a telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium, and comprising a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call, the method comprising acts of:

accessing the telephony application on the telephony communication device after the telephony communication device is deployed on the communications network; and

modifying the telephony application with modifications developed independently from creation of the telephony software component.

22. A communications network comprising:

a transmission medium; and

one or more telephony communication devices, each telephony communication device comprising

a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium; and

a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call,

wherein the telephony application is modifiable after the telephony communication device is deployed on the communications network with modifications developed independently from creation of the telephony software component.

23. A first telephony communication device that is part of a communications network that comprises a transmission medium, the first telephony communication device comprising:

a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium; and

a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call,

wherein at least a part of an additional telephony application can be added to the telephony software component after the first telephony communication device is deployed on the communications network.

24. The first telephony communication device of claim 23, wherein the additional telephony applications was developed independently from creation of the software component.

25. The first telephony communication device of claim 23, wherein the telephony software component further comprises an open application programming interface to add the at least part of the additional telephony application.

26. The first telephony communication device of claim 23, wherein the telephony software component further comprises:

at least a part of a non-telephony application defining a non-telephony function to be performed independently of a telephone call.

27. The first telephony communication device of claim 23, wherein the telephony application is modifiable after the first telephony communication device is deployed on the communications network with modifications developed independently from creation of the telephony software component.

28. The first telephony communication device of claim 23, wherein the telephony application is written in a general purpose programming language.

29. The first telephony communication device of claim 23, wherein the general purpose programming language is Java.

30. The first telephony communication device of claim 23, wherein the telephony software component further comprises:

an operating system; and

an operating system interface to interface communications between the operating system and the telephony applications of the telephony software component such that the telephony applications are independent of the operating system.

31. The first telephony communication device of claim 23, wherein the telephony software component further comprises:

a call processing module to represent and control concurrently one or more telephone calls, each telephone call including one or more connections to other users corresponding to other telephony communication devices on the communications

network, wherein, for each telephone call, the call processing module represents and controls each connection.

32. The first telephony communication device claim 31, wherein, for each connection, the call processing module is operative to control communications on the telephony device corresponding to the connection using a first call control protocol selectable from a plurality of call control protocols available on the first telephony communications device.

33. The first telephony communication device of claim 32, wherein the telephony software component further comprises:

an application programming interface to develop applications for the first telephony communication device, wherein the application programming interface is generic to the plurality of call control protocols.

34. The first telephony communication device of claim 31, wherein the telephony software component further comprises:

a media processing module, the media processing module comprising, for each telephone call, a corresponding media processing component to represent media processing of the telephone call, and for each connection of the telephone call, the media processing component includes a corresponding media processing connection,

wherein, for each telephone call, to control the media processing of the telephone call, the call processing module controls the media processing component corresponding to the telephone call, and, for each connection of the telephone call, to control the media processing of the connection, the call processing module controls the corresponding media processing connection.

35. The first telephony communication device of claim 31, wherein the call processing module is operative to communicate with one or more other network resources on the communications network to control the one or more telephone calls.

36. The first telephony communication device of claim 35, wherein a second telephony communication device is connected to the communications network through a Public Switched Telephone Network and an IP/PSTN gateway, and

wherein the call processing module is operative to communicate with the IP/PSTN gateway to control, during the telephone call, communications between the first user and a second user of the second telephony communication device.

37. The first telephony communication device of claim 23, wherein the telephony software component further comprises:

a media processing module, the media processing module comprising media processing elements to define media processing functions;

wherein, to implement a media processing function, the telephony application is operative to dynamically configure one or more combinations of the media processing elements during execution of the telephony application.

38. The first telephony communication device of claim 23, wherein the telephony software component further comprises a pointer to a location on the communications network of an application remotely executable by the first telephony communication device.

39. The first telephony communication device of claim 23, wherein the telephony application defines a remote user interface for another network resource on the communications network.

40. The first telephony communication device of claim 23, wherein the first telephony communication device is a telephone.

41. A method of defining functionality for a telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium, and comprising a telephony software component to control

operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call, the method comprising acts of:

after the telephony communication device is deployed on the communications network, accessing the telephony software component on the telephony communication device; and

adding at least a part of an additional telephony application to the telephony software component.

42. A system for defining functionality for a telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium, and comprising a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call, the system comprising:

means for accessing the telephony software component after the telephony communication device is deployed on the communications network; and

means for adding at least a part of an additional telephony application to the telephony software component.

43. A computer program product, comprising:

a computer-readable medium; and

computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a computer, instruct the computer to perform a method of defining functionality for a telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to

the transmission medium, and comprising a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call, the method comprising acts of:

accessing the telephony software component after the telephony communication device is deployed on the communications network; and

adding at least a part of an additional telephony application to the telephony software component.

44. A communications network comprising:

a transmission medium; and

one or more telephony communication devices, each telephony communication device comprising

a telephony hardware component comprising one or more inputs to receive audio input from a first user and first data from the transmission medium, and one or more outputs to transmit media to the first user and second data to the transmission medium; and

a telephony software component to control operation of the hardware component, the software component comprising at least a portion a telephony application defining a telephony function to be performed in association with a telephone call,

wherein at least a part of an additional telephone applications can be added to the telephony software component after the telephony communication device is deployed on the communications network.

45. A telephony communication device that is part of a communications network that comprises a transmission medium, the telephony communication device comprising:

a call processing module to represent and control at least a first telephone call that includes one or more connections to other users, each user corresponding to another telephony communication device on the communications network,

wherein, for at least a first of the one or more connections, the call processing module is operative to control communications corresponding to the first connection on the telephony communication device using a first call control

protocol selectable from a plurality of call control protocols available on the first telephony communication device.

46. The telephony communication device of claim 45, wherein, concurrently to controlling the communications corresponding to the first connection, the call processing module is operative to control communications on the telephony device corresponding to a second of the one or more connections using a second of the plurality of call control protocols.

47. The telephony communications device of claim 45, wherein the call processing module represents and controls a second telephone call that represents at least a second connection,

wherein, concurrently to controlling communications corresponding to the first connection, the call processing module is operative to control communications corresponding to the second connection on the telephony communication device using a second of the plurality of call control protocols.

48. The telephony communication device of claim 45, wherein the first call control protocol is the Session Initiation Protocol.

49. The telephony communication device of claim 45, wherein the first call control protocol is the first telephone call using the H.323 protocol.

50. The telephony communication device of claim 45, wherein the first call control protocol is the Media Gateway Control Protocol.

51. The telephony communication device of claim 45, wherein the call processing module is operative to control one or more of the connections of the first telephone call using the Megaco/H.248 Protocol

52. The telephony communication device of claim 45, wherein the call processing module is operative to control one or more of the connections of the first telephone call using the Skinny Station Protocol.

53. A method of controlling one or more telephone calls on a first telephony communication device connected to a communications network, a first telephone call including one or more connections to other users, each user corresponding to another telephony communication device on the communications network, the method comprising, for at least a first of the one or more connections, acts of:

selecting a first call control protocol from a plurality of call control protocols available on the first telephony communication device; and

controlling communications corresponding to the first connection on the telephony communication device using the first call control protocol.

54. The method of claim 53, the method further comprising an act of:

concurrently to controlling the communications corresponding to the first connection, controlling communications on the telephony device corresponding to a second of the one or more connections using a second of the plurality of call control protocols,.

55. The method of claim 53, wherein a second telephone call includes at least a second connection, the method further comprising:

concurrently to controlling communications corresponding to the first connection, controlling communications corresponding to the second connection on the telephony communication device using a second of the plurality of call control protocols.

56. A system for controlling a first telephone call on a first telephony communication device connected to a communications network, the telephone call including one or more connections to other users, each user corresponding to another telephony communication device on the communications network, the system comprising, for at least a first of the one or more connections:

means for selecting a first call control protocol from a plurality of call control protocols available on the first telephony communication device; and

means for controlling communications corresponding to the first connection on the telephony communication device using the first call control protocol.

57. A computer program product, comprising:

a computer-readable medium; and

computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a computer, instruct the computer to perform a method of controlling a first telephone call on a first telephony communication device connected to a communications network, the telephone call including one or more connections to other users, each user corresponding to another telephony communication device on the communications network, the method comprising, for at least a first of the one or more connections, acts of:

selecting a first call control protocol from a plurality of call control protocols available on the first telephony communication device; and

controlling communications corresponding to the first connection on the telephony communication device using the first call control protocol.

58. A communications network comprising:

a transmission medium;

one or more telephony communication devices, each telephony communication device comprising

a call processing module to represent and control at least a first telephone call that includes one or more connections to other users, each user corresponding to another telephony communication device on the communications network,

wherein, for at least a first of the one or more connections, the call processing module is operative to control communications corresponding to the first connection on the telephony communication device using a first call control protocol selectable from a plurality of call control protocols available on the first telephony communication device.

59. A computer program product, comprising:

a computer-readable medium; and

computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a first telephony communication device, instruct the first telephony communication device to perform a telephony application, wherein at least a portion of the telephony application may be added to the first telephony communication device after the first telephony communication device has been deployed on a communications network.

60. The computer program product of claim 59, wherein the telephony application defines a method of indicating an incoming call to at least a first user of a second telephony communication device, the method comprising acts of:

receiving a dialing instruction corresponding to a network address of the second telephony communication device;

identifying the first user from the dialing instructions;

selecting a sound to play based on the identification; and

sending a call set-up message to the second telephony communication device at the network address, the call set-up message including information to cause the second telephony communication device to produce the selected sound.

61. The computer program product of claim 60, wherein the information includes a signal representing the selected sound.

62. The computer program product of claim 60, wherein the information includes an instruction to play the selected sound.

63. The computer program product of claim 60, wherein the information includes an instruction to play the selected sound from a particular location.

64. The computer program product of claim 59, wherein the telephony application defines a method of selectively transmitting media during a conference call that includes a first user input for receiving media from the first user and two or more connections between the first telephony communication device and two or more second telephony

communication devices, respectively, each second telephony communication device corresponding to a second user, the method comprising acts of:

receiving a user selection identifying one or more third users to exclude from receiving media from one or more fourth users, wherein each third and fourth user is either the first user or one of the second users,

during the conference call, receiving one or more media inputs from one or more of the fourth users; and

during the conference call, preventing the one or more third users from receiving any media produced from the one or more media inputs.

65. The computer program product of claim 64, wherein the first telephony communication device further comprises a local call bridge, and first application performs the act of preventing using the local call bridge.

66. The computer program product of claim 64, wherein the user selection is received from the first user.

67. The computer program product of claim 64, wherein the user selection is received from one of the second users.

68. The computer program product of claim 59, wherein the telephony application defines a method of selectively transmitting media during a conference call that includes a first user input for receiving media from the first user and two or more connections between the first telephony communication device and two or more second telephony communication devices, respectively, each second telephony communication device corresponding to a second user, the method comprising acts of:

receiving a user selection identifying one or more third users to include in receiving media from one or more fourth users, wherein each third and fourth user is either the first user or one of the second users,

during the conference call, receiving one or more media inputs from one or more of the fourth users; and

during the conference call, permitting the one or more third users to receive media produced from the one or more media inputs.

69. The computer program product of claim 68, The computer program product of claim 22, wherein the first telephony communication device further comprises a local call bridge, and first application performs the act of permitting using the local call bridge.

70. The computer program product of claim 68, wherein the user selection is received from the first user.

71. The computer program product of claim 68, wherein the user selection is received from one of the second users.

72. The computer program product of claim 59, wherein the telephony application defines a method of communicating, as part of a telephone call, textual information from a first user of a first telephony communication device to a second user of a second telephony communication device connected to the first telephony communication device by a first connection, wherein the first telephony communication device comprises a display screen, the method comprising acts of:

during the telephone call, receiving on the first connection from the second telephony communication device first textual data defining text; and

displaying on the display screen the text defined by the textual data.

73. The computer program product of claim 59, wherein the telephony application defines a method of performing a telephone call between a first user of the first telephony communication device and one or more second users of second telephony communication devices, the method comprising acts of:

receiving one or more user identifiers, each user identifier identifying one of the second users;

receiving date and time information specifying a future date and time to conduct the call;

storing the user identifiers and the data and time information;

at the date and time specified by the date and time information, accessing the user identifiers; and

initiating a telephone call between the first telephony communication device and the one or more second telephony communication devices.

74. The computer program product of claim 73, wherein the act of receiving comprises receiving two or more user identifiers, and wherein the act of initiating comprises initiating a conference call between the first telephony communication device and two or more second telephony communication devices.

75. The computer program product of claim 74, further comprising and act of:

receiving a conference bridge identifier identifying a conference bridge,

wherein the act of initiating the telephone call includes calling the conference bridge.

76. The computer program product of claim 73, further comprising:

before initiating the telephone call, notifying the one or more second users about the telephone call by sending a message including information about the telephone call from the first telephony communication device to the one or more second telephony communication devices.

77. The computer program product of claim 59, wherein the telephony application defines a method of initiating a telephone call between a first user of the first telephony communication device and one or more second users of second telephony communication devices, the method comprising acts of:

receiving one or more uniform resource locator selections made by the first user, each uniform resource locator selection corresponding to one of the second telephony communication devices; and

initiating, from the first telephony communication device, a telephone call between the first telephony communication device and the one or more second telephony communication devices.

78. The computer program product of claim 77, wherein the first application includes a web page defined on the first telephony communication device, and the method further comprising acts of:

receiving one or more user tag selections, each tag selection indicating a tag selected using a web browser from the web page; and

determining the one or more uniform resource locators from the one or more selected tags.

79. The computer program product of claim 77, wherein the first telephony communication device includes an embedded web server, and the method further comprising acts of:

providing the first user access to the web server after the first telephony communication device has been deployed on the communications network;

receiving instructions from the first user entered through the web server; and

creating the web page from the instructions.

80. The computer program product of claim 59, wherein the telephony application defines a method of displaying text associated with media received on a first connection between the first telephony communication device and a second telephony communication device as part of a telephone call, and wherein the first telephony communication device further comprises a display screen, the method comprising acts of:

receiving on the first connection a media signal from the second telephony communication device, the media signal representing media content;

before or concurrently to receiving the media signal, receiving on the first connection a textual signal representing text content corresponding to the media content;

displaying the text content on the display screen; and

after or concurrently to displaying the text content, playing the media content.

81. The computer program product of claim 80, wherein the textual signal is received in response to sending an indication to the second telephony communication device that the first telephony communication device is capable of displaying textual data corresponding to a telephone call.

82. The computer program product of claim 80, wherein the media content comprises a voice produced by an interactive voice response system, the voice speaking one or

more options, and wherein the textual content textually represents the one or more options.

83. The computer program product of claim 59, wherein the telephony application defines a method of communicating information in accordance with a state of a second telephony communication device, wherein the first further comprises a display screen, the method comprising acts of:

sending a call set-up message to the second telephony communication device to set-up the telephone call;

in response to the call set-up message, receiving a textual signal from the second telephony communication device, the textual signal defining text content corresponding to a state of the second telephony communication device; and

displaying the text content on the display screen of the first telephony communication device.

84. The computer program product of claim 83, wherein the text content comprises one or more options corresponding to the state of the second telephony communication device, each option indicating an action to be taken in response to the state of the second communication device, and wherein displaying the text content comprises displaying the one or more options to the user, the method further comprising acts of;

receiving a selection of a first of the one or more options from the user; and

sending a selection signal representing the selection to the second communication device to perform the action indicated by the selection.

85. The computer program product of claim 59, wherein the telephony application defines a method of communicating information in accordance with a state of the first telephony communication device to a second telephony communication device, the method comprising acts of:

receiving, at the first telephony communication device, a call set-up message from the second telephony communication device to initiate a telephone call;

determining a state of the first telephony communication device; and

transmitting a textual signal representing the determined state from the first telephony communication device to the second telephony communication device for display.

86. The computer program product of claim 85, wherein the textual signal comprises textual representations of one or more options for a user of the second telephony communication device to be displayed on the second telephony communication device, each option indicating an action to be taken in response to the state of the second communication device, the method further comprising acts of:

receiving a selection signal from the second communication device, the selection signal representing a selection of one of the options by a user to the second communication device; and

performing the action indicated by the selection.

87. The computer program product of claim 59, wherein the telephony application defines a method of communicating an audible expression during a telephone call, the method comprising acts of:

receiving a voice signal representing a voice of a first user of the first telephony communication device;

receiving a selection signal representing a selection of an audible expression by the first user;

generating the audible expression signal corresponding to the selection signal, the audible expression signal representing the audible expression;

mixing the audible expression signal and the voice signal to produce a mixed signal;

transmitting the mixed signal, as part of the telephone call, to one or more second users at one or more second telephony communication devices.

88. The computer program product of claim 87, wherein, the first telephony communication device further comprises a display screen, and the method further comprises an act of:

displaying one or more icons on the display screen, each icon representing an audible expression,

wherein the selection signal is produced by the first user selecting an icon from the display screen.

89. The computer program product of claim 59, wherein the telephony application defines a method of screening a telephone call based on information associated with the call, the method comprising acts of:

receiving at the first telephony communication device a first call set-up message from a second telephony communication device;

accessing one or more call answering rules, each call answering rule defining a condition and an action to be taken by the first telephony communication device in response to receiving a call set-up message if the condition is satisfied;

for one or more of the call answering rules, determining whether the condition is satisfied; and

responding to the call set-up message in accordance with the determinations for the one or more call answering rules.

90. The computer program product of claim 89, wherein a first of the one or more rules defines an action to be taken on condition of a call set-up message being received during one or more time intervals,

wherein the act of determining comprises

determining a first time at which the first call set-up is received; and

comparing the first time to the one or more time intervals to determine whether the first time is within one or more of the time intervals; and

wherein, if the first time is within the one or more time intervals, the act of responding to the call set-up message comprises responding to the call set-up message in accordance with the action defined by the first rule.

91. The computer program product of claim 89, a first of the one or more rules defines an action to be taken on condition of the call set-up message being from one or more particular users,

wherein the act of receiving comprises receiving an identification signal corresponding to the call set-up message, the identification signal identifying a first user of the second communication device;

wherein the act of determining comprises

identifying the first user from the identification signal; and

comparing the identification of the first user to identifications of one or more particular users to determine if the first user is one of the particular users, and

wherein, if the first user is one of the particular users, the act of responding to the call set-up message comprises responding to the call set-up message in accordance with the action defined by the first rule.

92. The computer program product of claim 89, wherein a first rule of the one or more call answering rules defines forwarding the call set-up message to another first telephony communication device if a first condition is satisfied,

wherein the act of responding comprises forwarding the call set-up message to the other first telephony communication device if the first condition is satisfied.

93. The first telephony communication device of 89, wherein a first call answering rule defines playing a sound if a first condition is satisfied,

wherein the act of responding comprises playing the sound if the first condition is satisfied.

94. The computer program product of claim 85, wherein a first call answering rule defines prompting a user of the second telephony communication device for more information if a first condition is satisfied,

wherein the act of responding comprises prompting the user for more information if the first condition is satisfied.

95. The computer program product of claim 59, wherein the telephony application defines a method of playing textual content of an e-mail message as audio on the first telephony communication device, the method comprising acts of:

controlling the electronic email message text to be sent to a network resource that converts the textual content of the e-mail message to a voice signal;

setting up a telephone call between the network resource and the first telephony communication device;

controlling transmitting the voice signal to the first telephony communication device as part of the telephone call; and

playing the voice signal on the first telephony communication device.

96. The computer program product of claim 59, wherein the telephony application defines a method of communicating, to a second user of at least a first part of a telephone call between a first user of the first telephony communication device and one or more third users of one or more third first telephony communication devices, the method comprising acts of:

recording as an audio file at least the first part of the telephone call between the first user and the one or more third users;

sending the audio file to an audio playing device corresponding to the second user.

97. The computer program product of claim 96, wherein the act of sending comprises attaching the audio file to an e-mail message and sending an e-mail message to the audio playing device.

98. The computer program product of claim 96, wherein the act of sending comprises attaching the audio file to a call set-up message and sending the call set-up message to the one or more second telephony communication devices.

99. The computer program product of claim 59, wherein the telephony application defines a method of sending a graphical message to a first user of a second telephony communication device, the method comprising:

sending a call set-up message from the first telephony communication device to the second telephony communication device;

receiving an indication that the first user is not answering the call set-up message; and

sending a graphical file to the second telephony communication device to store as a message to the first user.

100. The computer program product of claim 59, wherein the telephony application defines a method of communicating a graphical message to a first user of the first telephony communication device, wherein the first telephony communication device includes a display screen, the method comprising:

receiving a call set-up message from a second telephony communication;

sending an indication that the first user is not answering the call set-up message;

in response to the sending of the indication, receiving as a message a graphical file from the second telephony communication device; and

displaying the graphical file on the display screen to communicate the message to the first user to store as a message to the first user.

101. The computer program product of claim 100, wherein the method further comprises an act of:

storing the graphical file on a storage medium accessible by the first telephony communication device.

102. The computer program product of claim 59, wherein the telephony application defines a method of placing one or more connections on hold during a telephone call, the method comprising acts of:

receiving an instruction to place a telephone call on hold;

receiving an indication of a location of a media file;

creating a connection to the location of the media file;

receiving media signals from the media file; and

sending the media signals to the one or more connections on hold.

103. The computer program product of claim 102, wherein the media file comprises audio data to be played at the one or more connections.

104. The computer program product of claim 102, wherein the media file comprises text data, graphical data or a combination thereof, and the act of playing comprises displaying the text data, graphical data or combination thereof on a display screen under control of the first telephony communication device.

105. The computer program product of claim 59, wherein the application defines a method of dynamically configuring a first telephony communication device for a first user, the method comprising acts of:

receiving a user identification identifying the first user;

accessing configuration information specific to the first user; and

configuring the first telephony communication device in accordance with the configuration information.

106. The computer program product of claim 59, wherein the telephony application defines a method of resolving conflicts between a plurality of applications on the first telephony communications device, the method comprising:

receiving an indication that a first application and one or more second applications are defined to be notified if a first event occurs; and

assigning notification priority to the first application such that, if first the event occurs, the first application is notified of the event and one or more second applications are not notified of the event.

107. The computer program product of claim 106, wherein the method further comprises acts of:

receiving an instruction from a user to assign notification priority to the first application, wherein the act of assigning is performed in response to receiving the instruction.

108. The computer program product of claim 59, wherein the telephony application defines, for a telephone call including a first user of the first telephony communication device and a plurality of second users of other telephony communication devices, a method of visually indicating to the first user that the voices of one or more third users of the plurality of second users are currently being played on the first telephony communication device, the method comprising:

displaying a plurality of user identifiers, each user identifier indicating one of the second users;

receiving audio data from one or more of the second users;

for each of the one or more third users, detecting voice data in the audio data received from the third user;

mixing the audio data of the third user to produce mixed audio data;

playing the mixed audio data on the first telephony communication device; and

for each of the one or more third users, displaying a visual indicator next to the identifier identifying the third user to indicate that the mixed audio data includes voice data from the third user.

109. A first telephony communication device, comprising:

a computer-readable medium; and

computer-readable signals stored on the computer-readable medium that define instructions that, as a result of being executed by a first telephony communication device, instruct the first telephony communication device to perform a telephony application, wherein at least a portion of the telephony application may be modified after the first telephony communication device has been deployed on a communications network with modifications developed independently from creation of the telephony application.

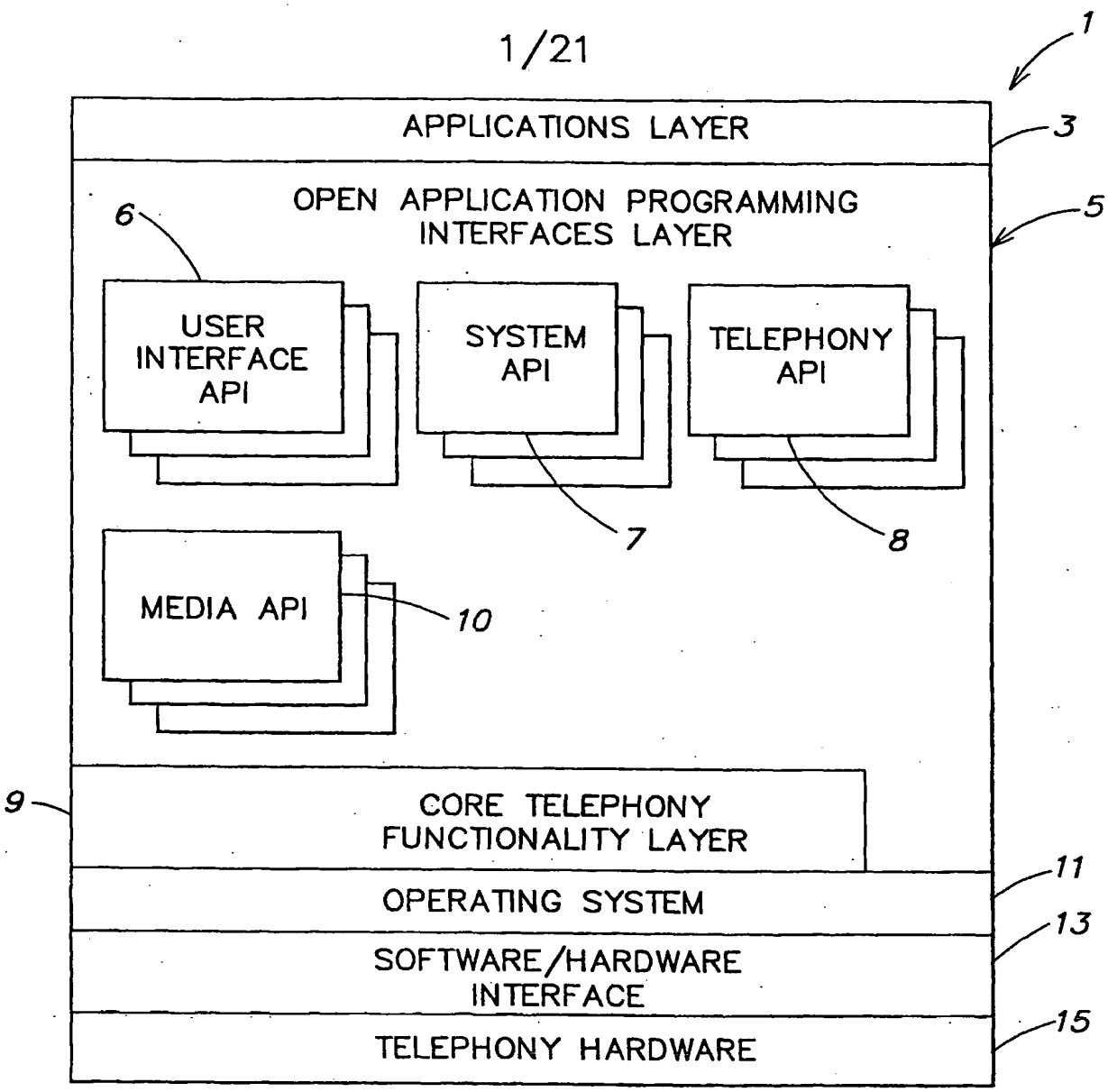


FIG. 1

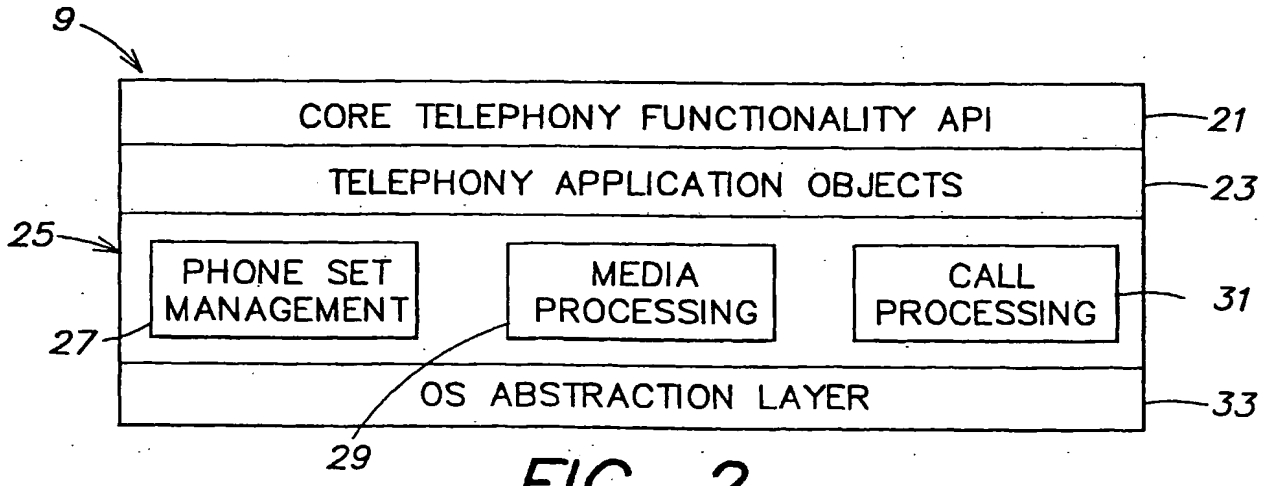


FIG. 2

BNSDOCID: <WO_____0131900A2_I_>

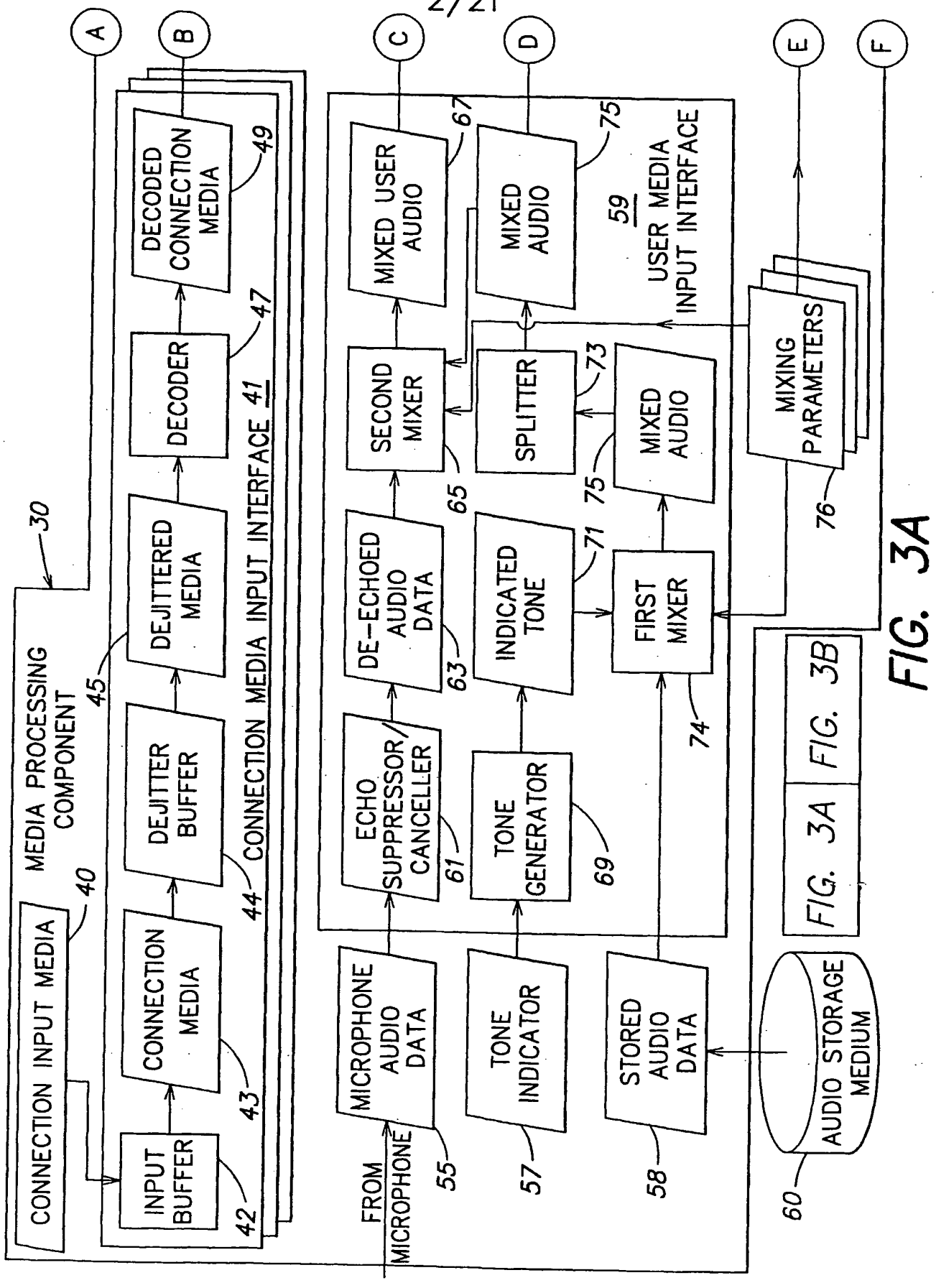


FIG. 3A

BNSDOCID: <WO_____0131900A2_I_>

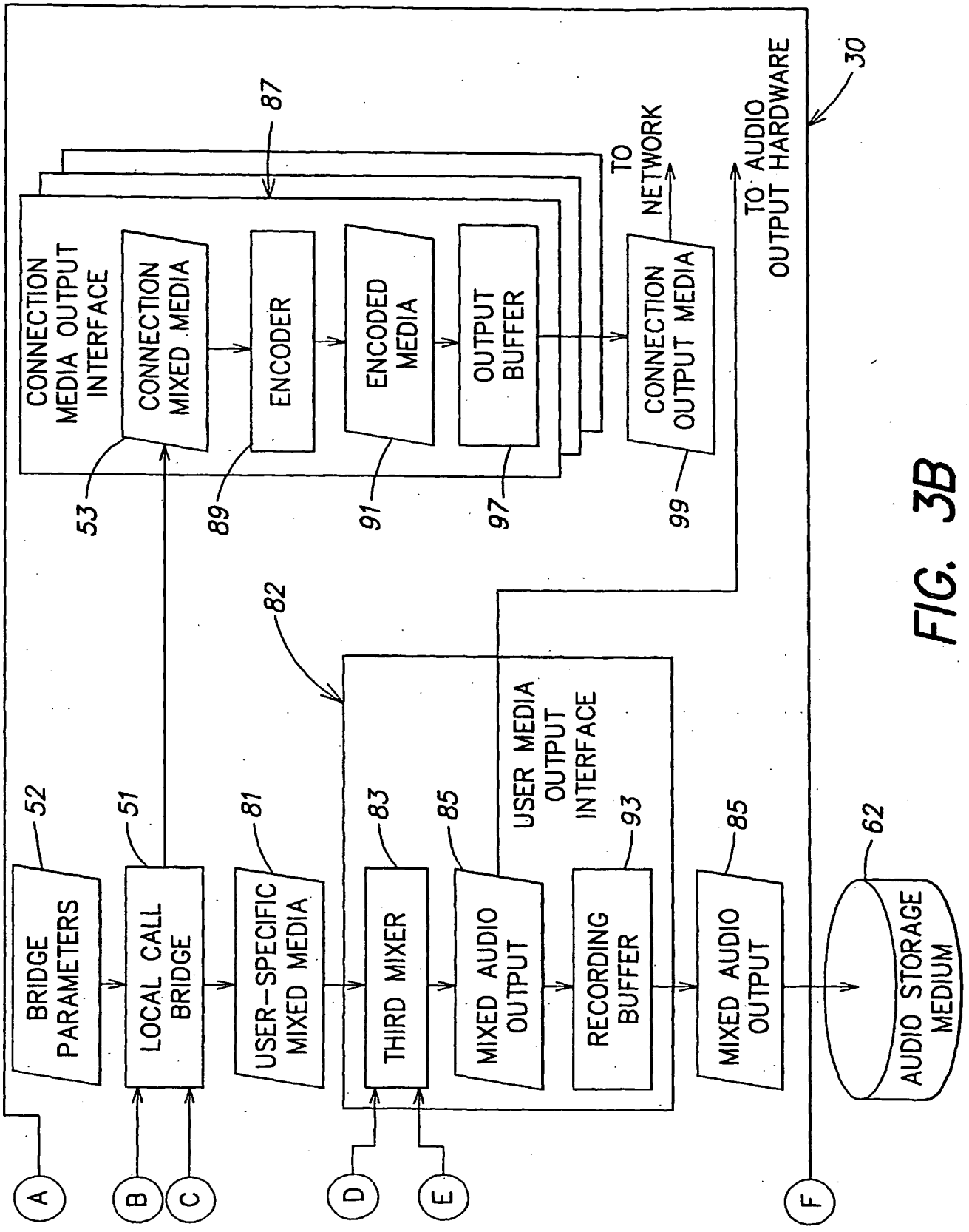


FIG. 3B

BNSDOCID: <WO_____0131900A2_I_>

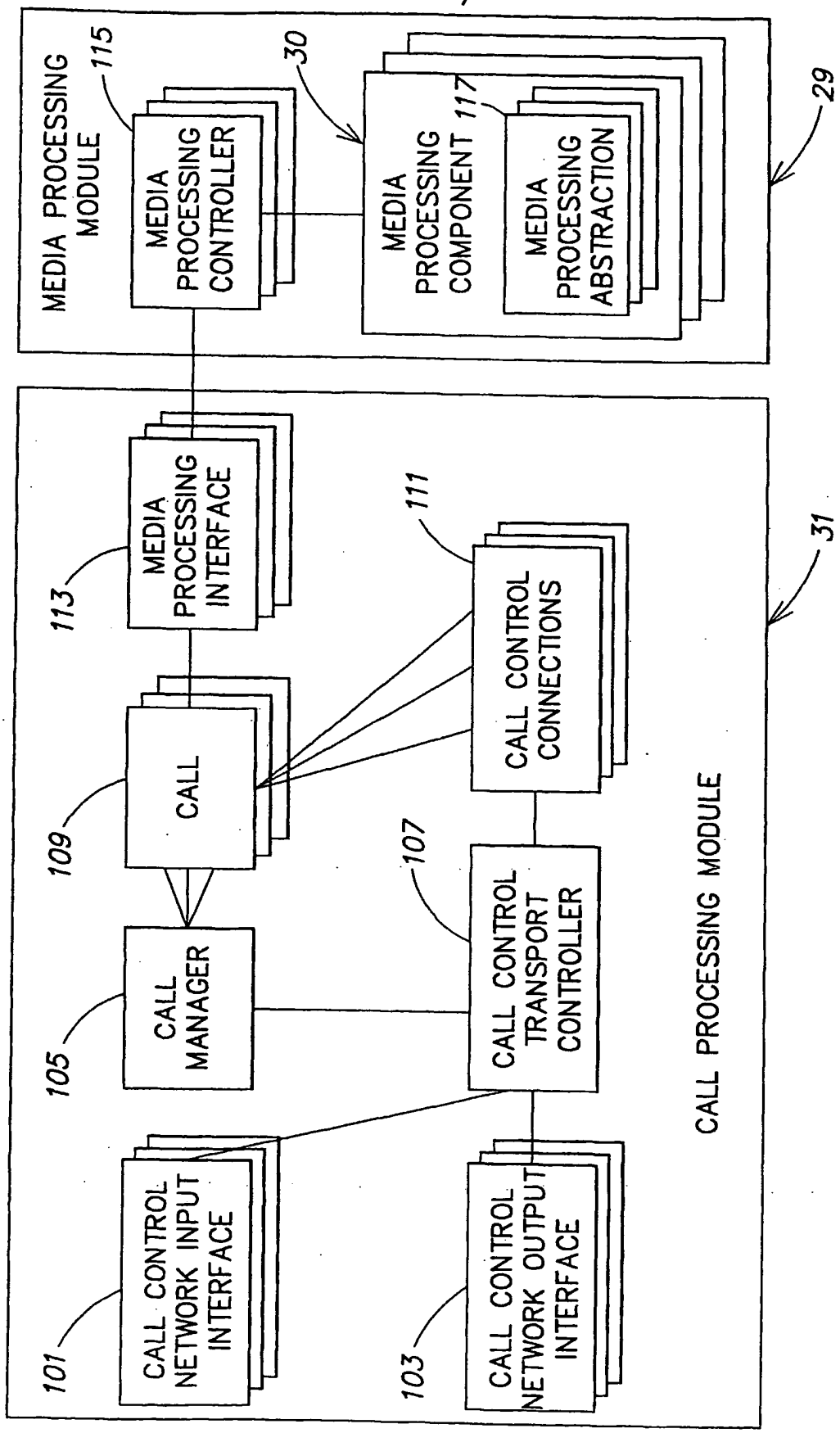


FIG. 4

BNSDOCID: <WO_____0131900A2_I_>

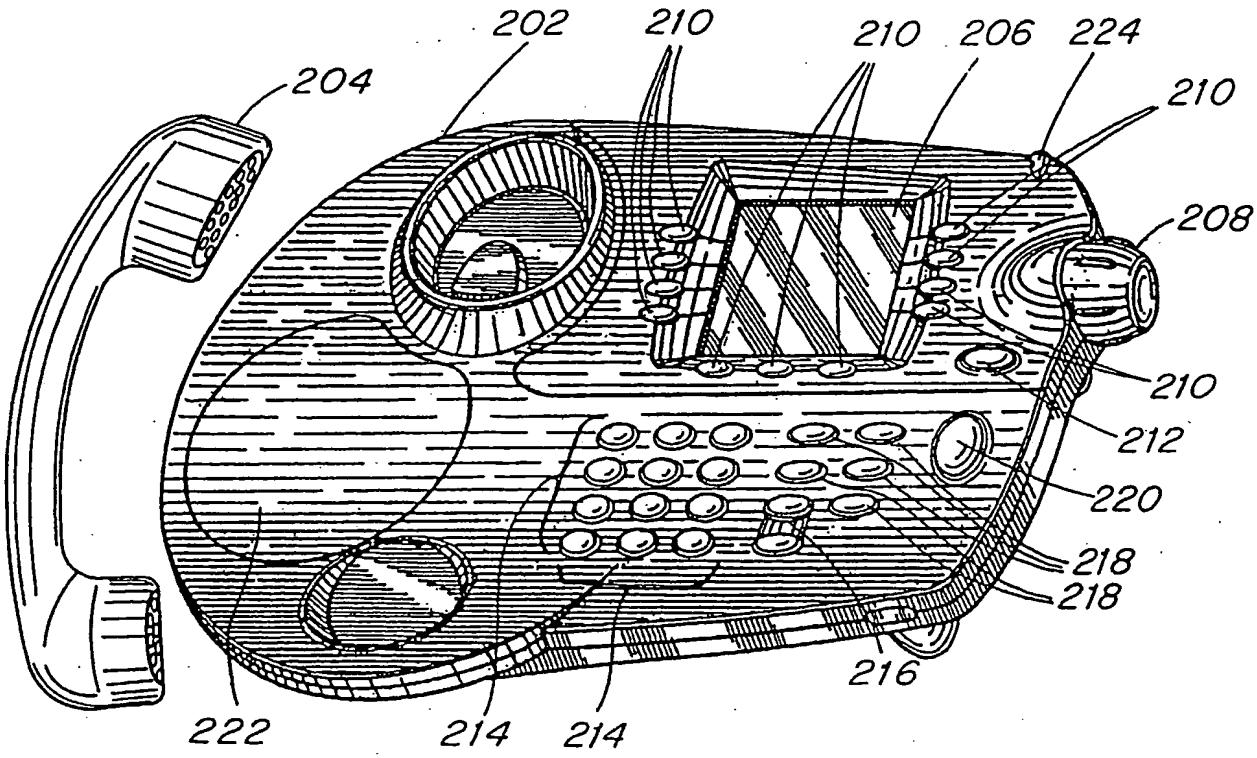


FIG. 5

BNSDOCID: <WO_____0131900A2_I_>

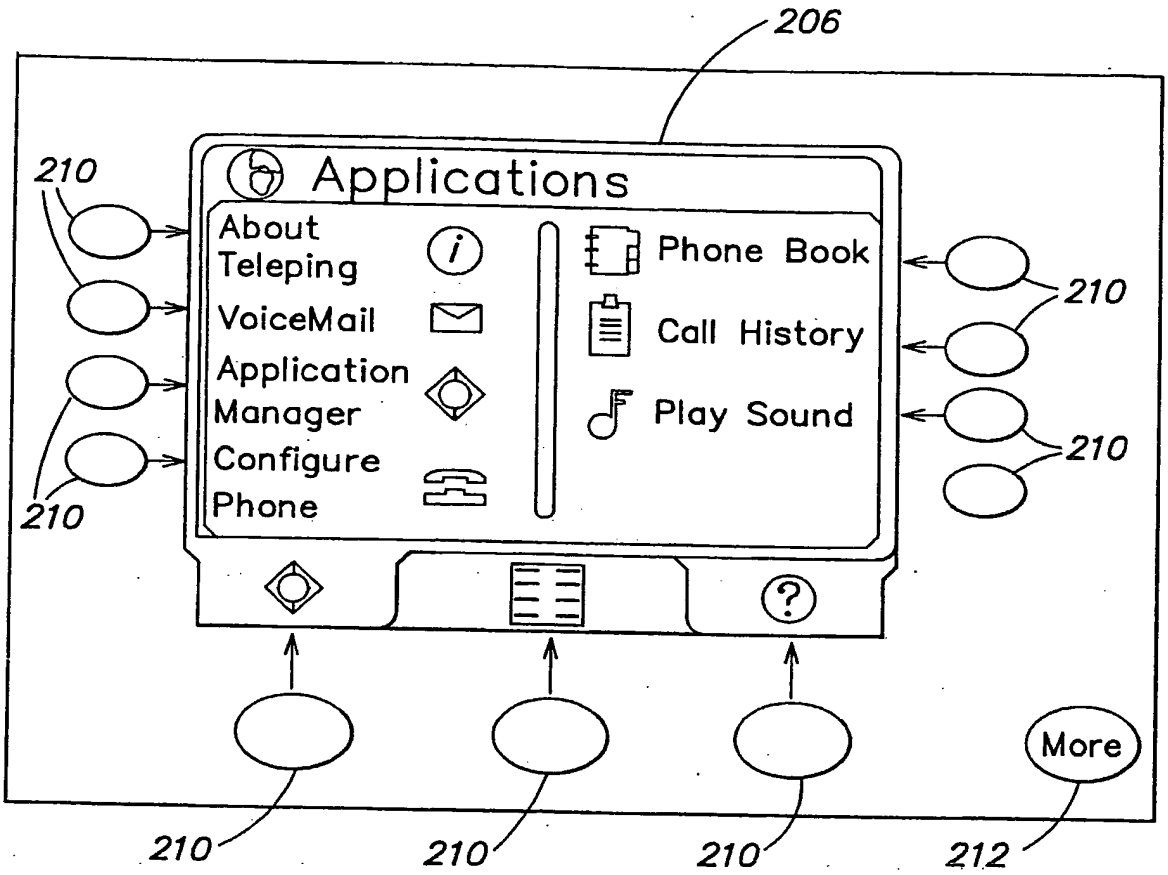


FIG. 6

BNSDOCID: <WO_____0131900A2_I_>

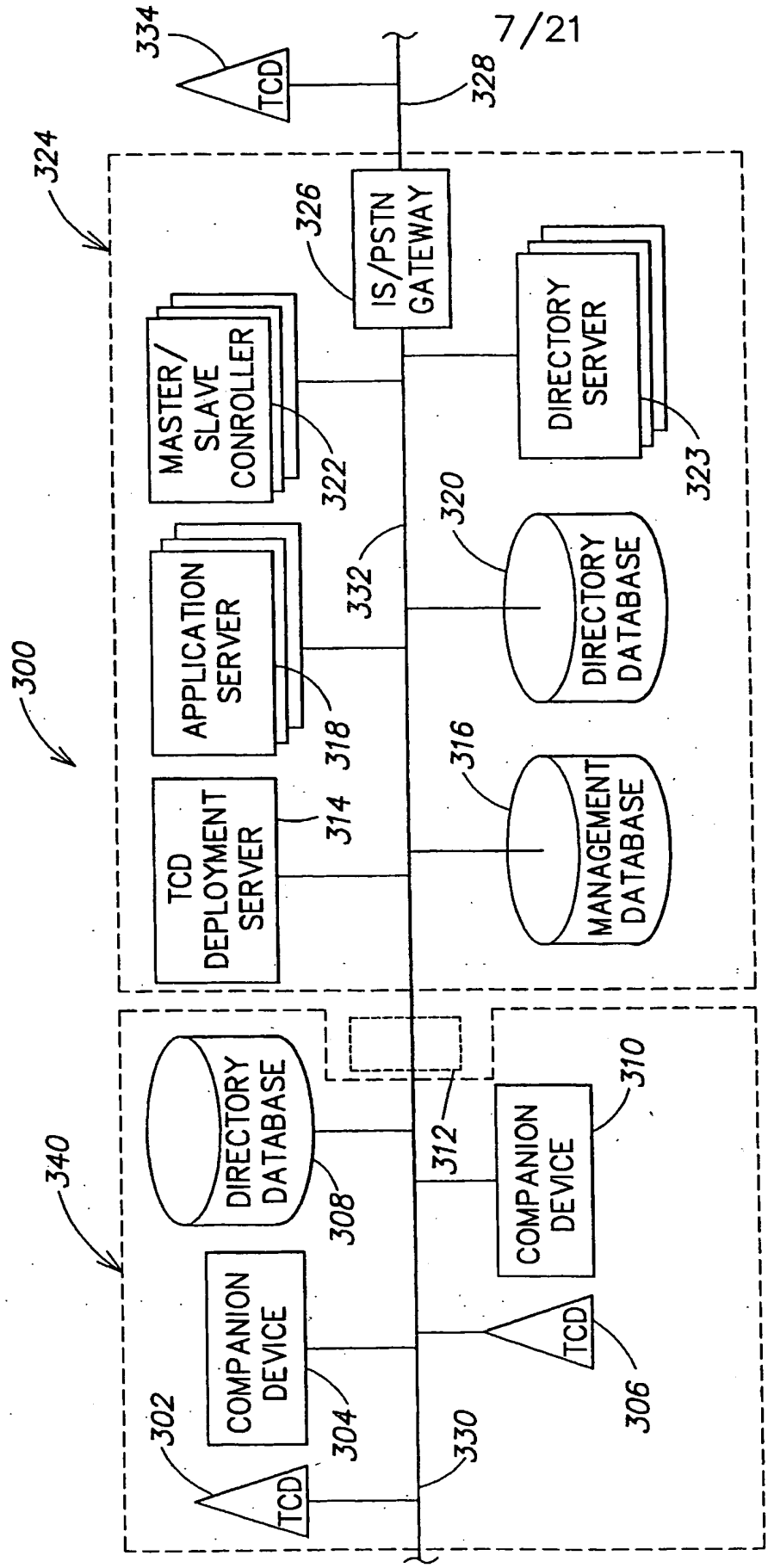


FIG. 7

BNSDOCID: <WO_____0131900A2_I_>

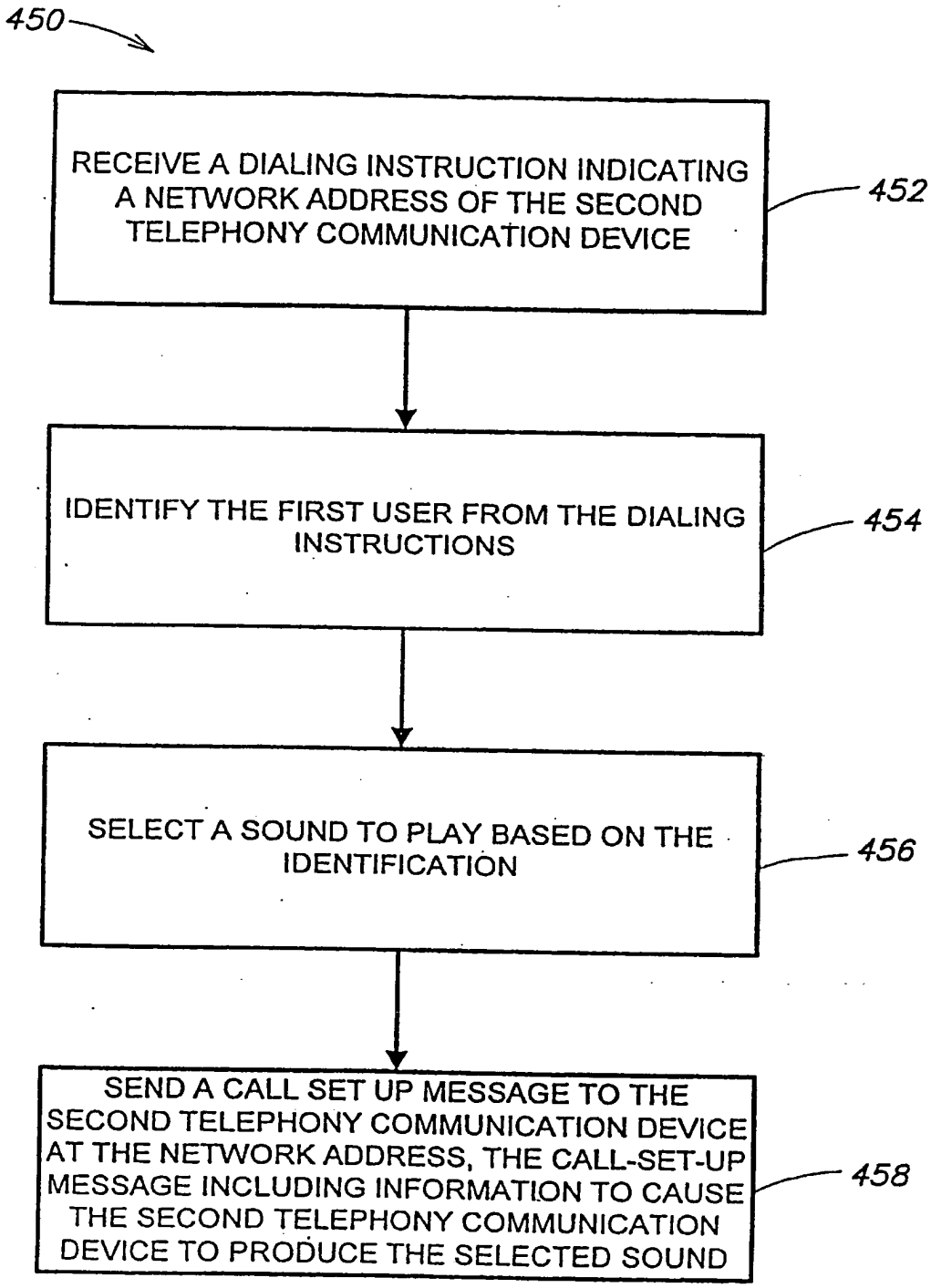


FIG. 8

BNSDOCID: <WO_____0131900A2_I_>

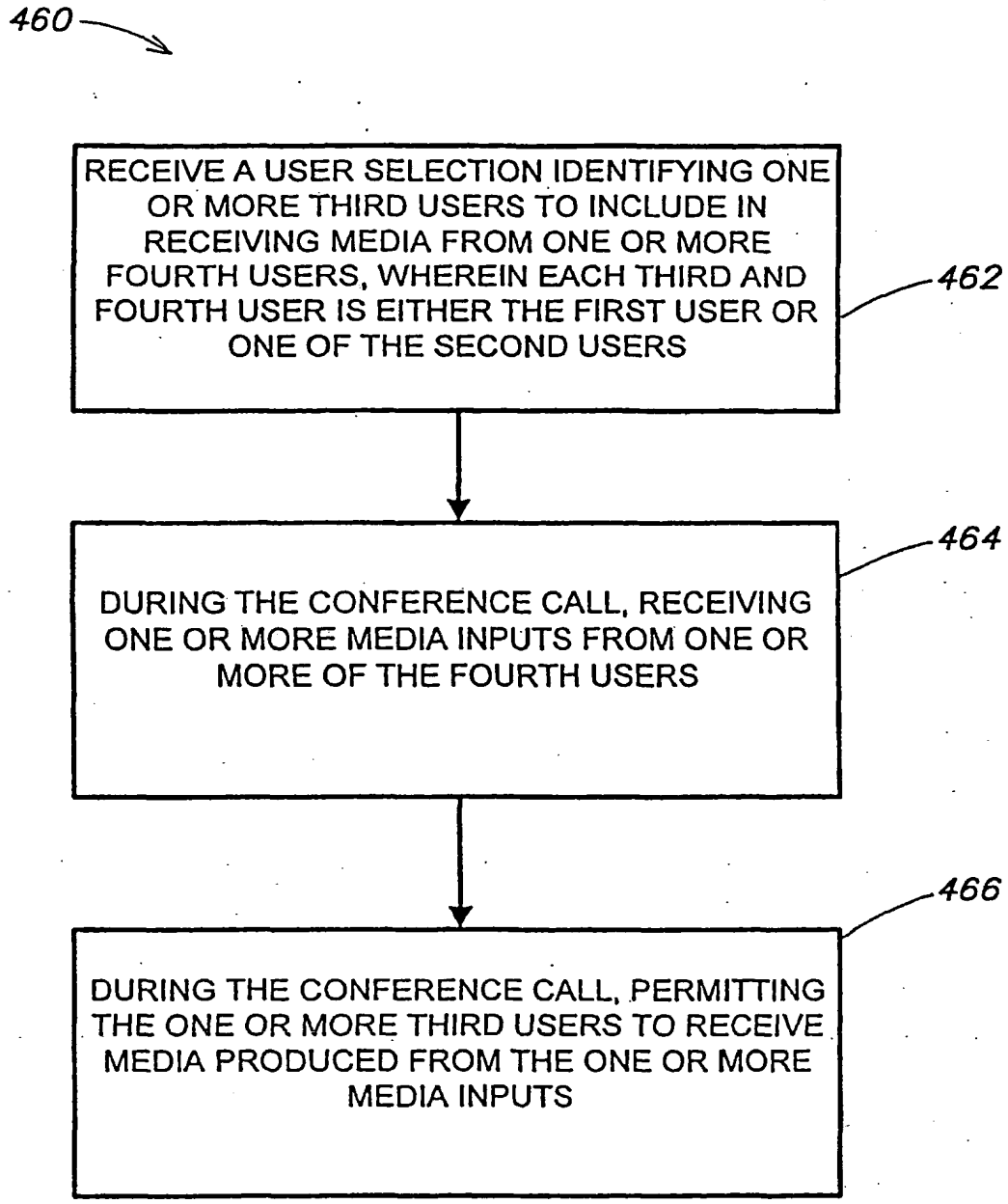


FIG. 9

BNSDOCID: <WO_____0131900A2_I_>

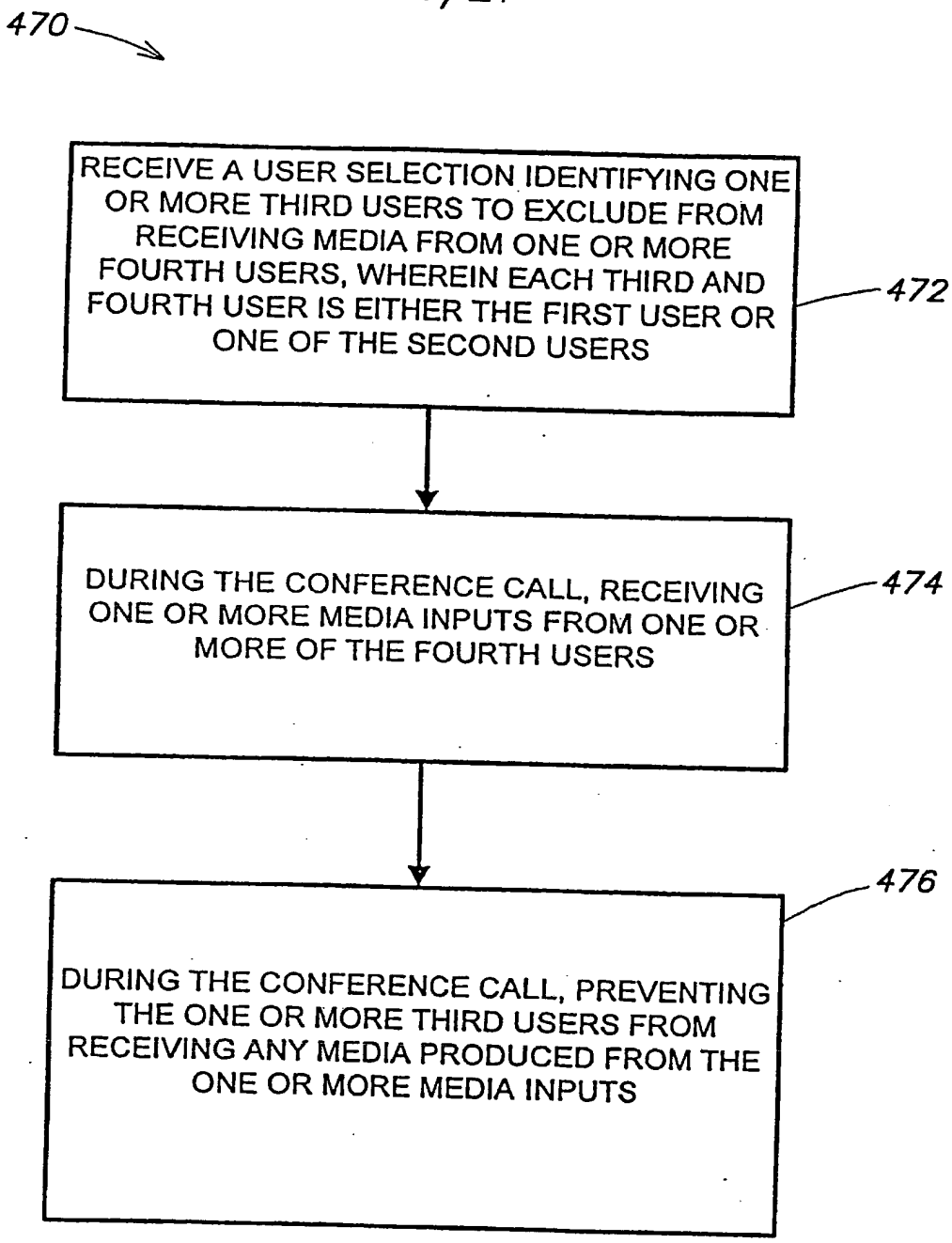


FIG. 10

BNSDOCID: <WO_____0131900A2_I_>

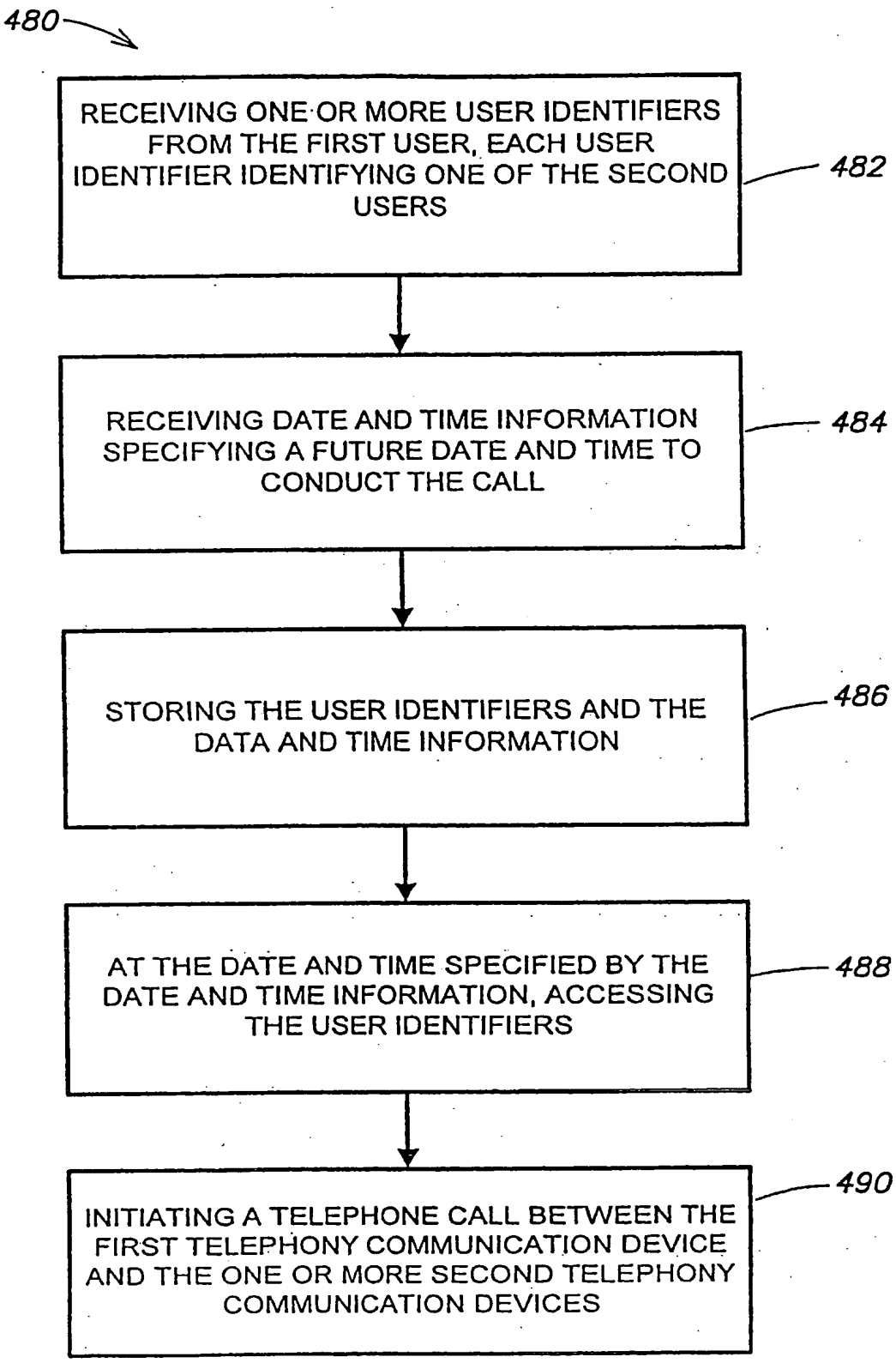


FIG. 11

BNSDOCID: <WO_____0131900A2_I_>

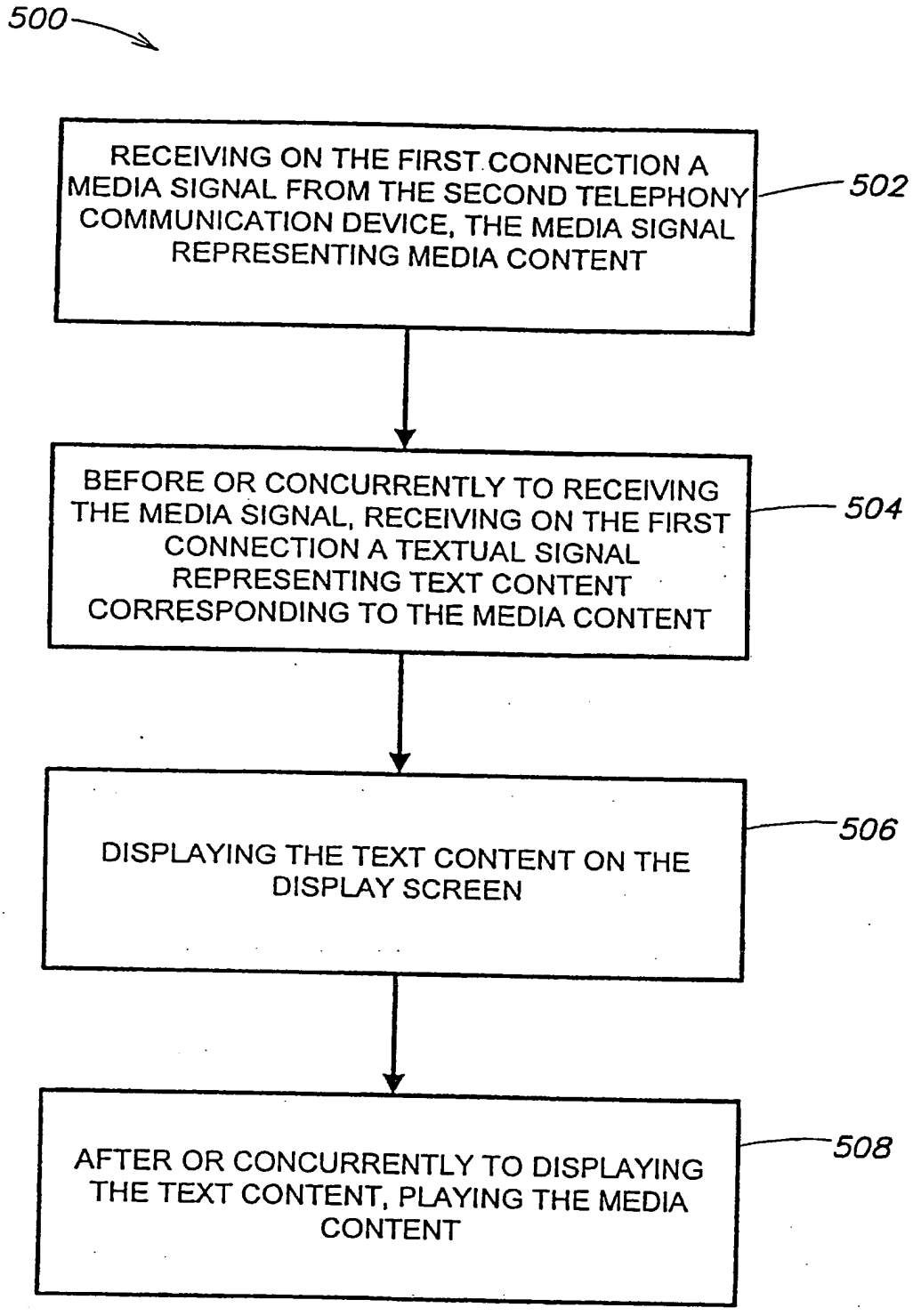


FIG. 12

BNSDOCID: <WO_____0131900A2_I_>

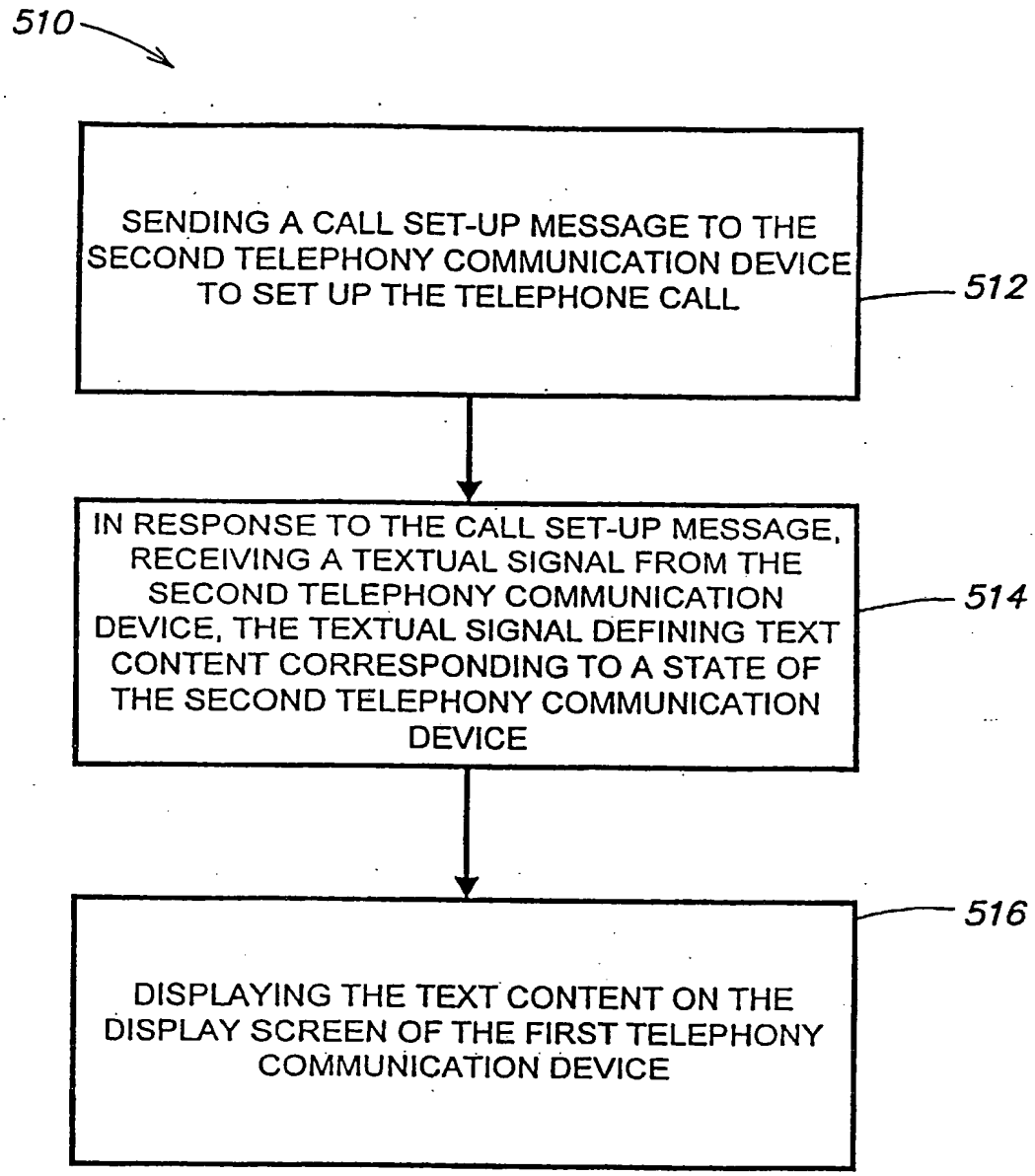


FIG. 13

BNSDOCID: <WO_____0131900A2_1_>

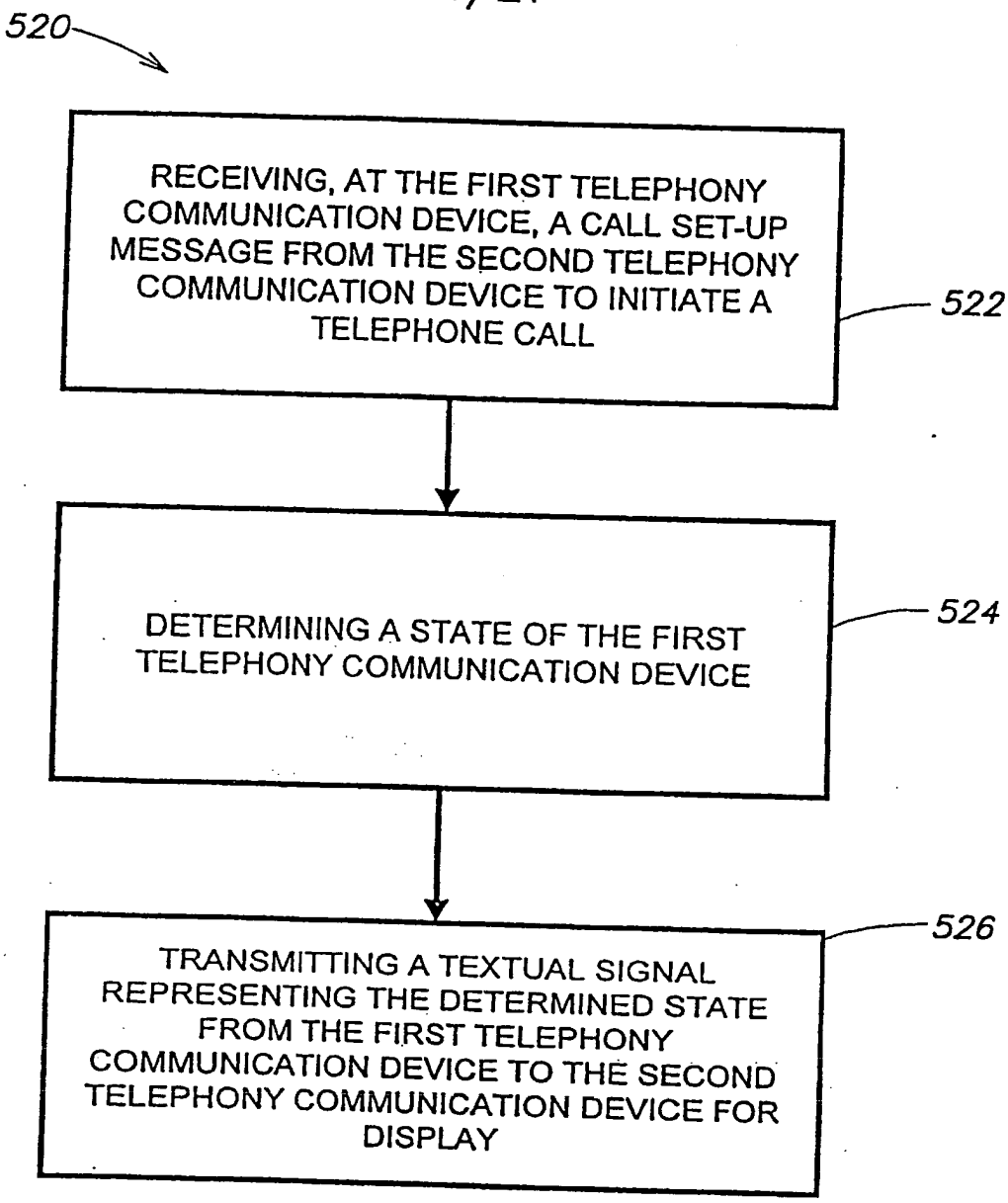


FIG. 14

BNSDOCID: <WO_____0131900A2_I_>

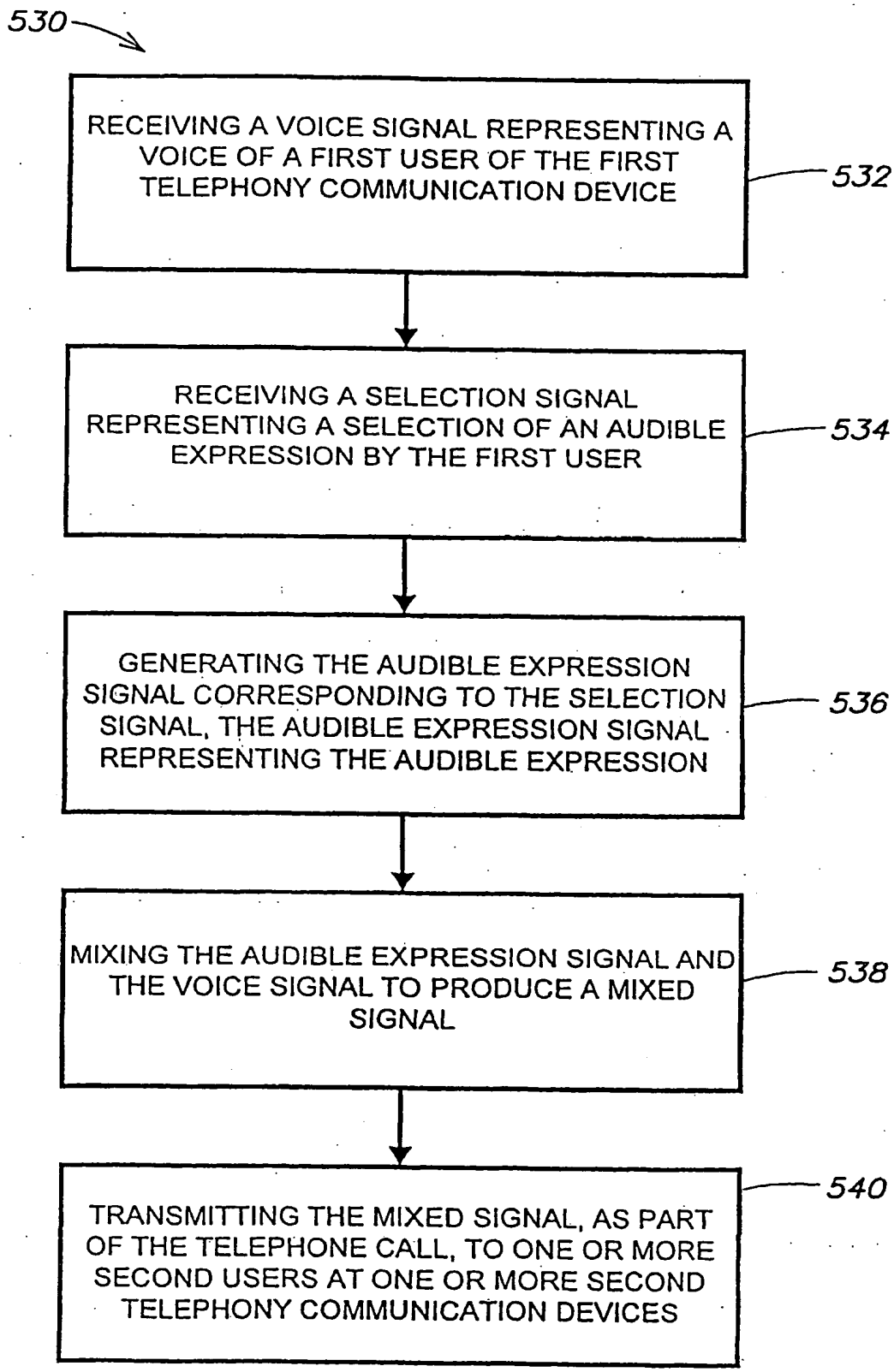


FIG. 15

BNSDOCID: <WO_____0131900A2_I_>

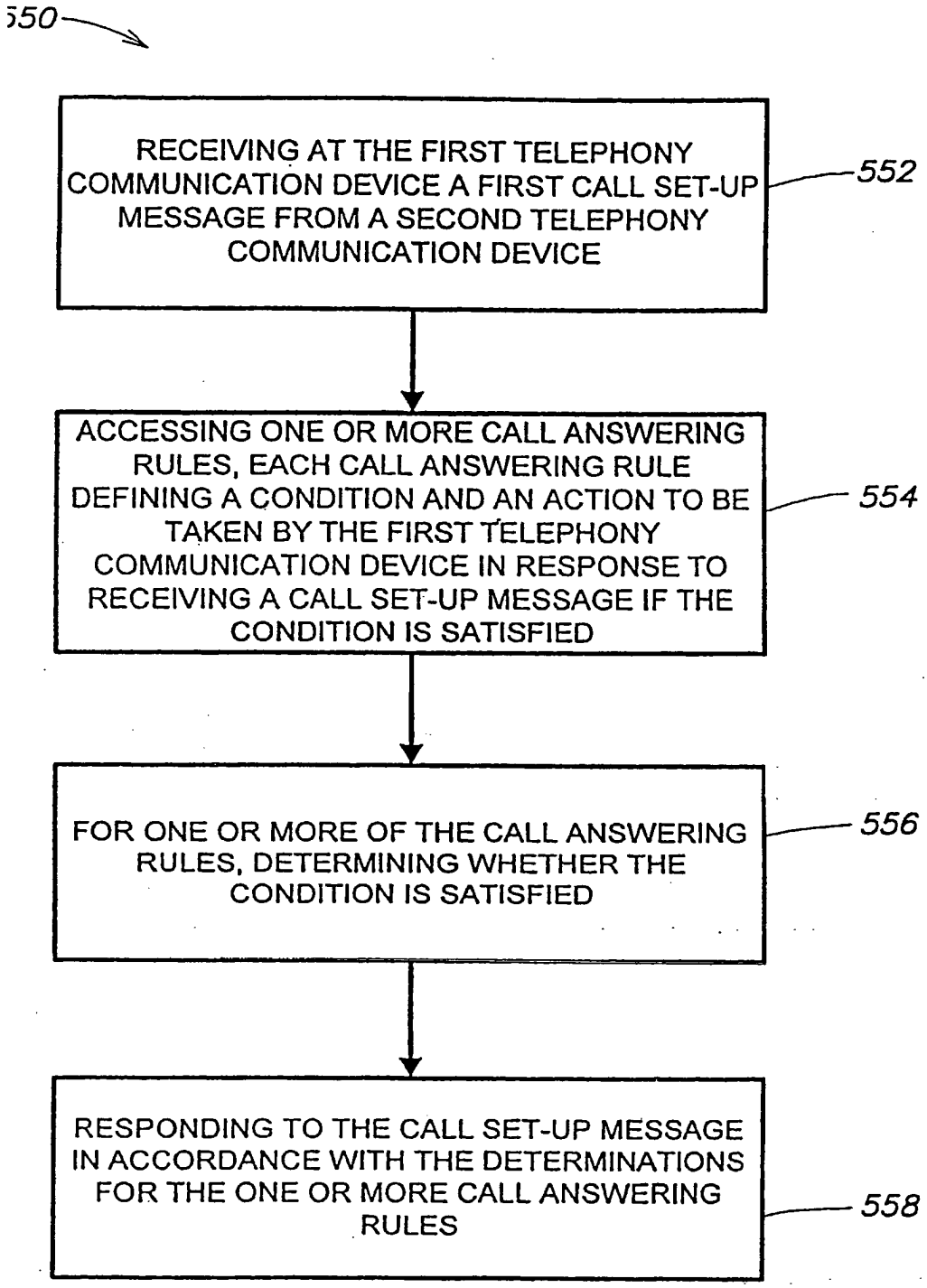


FIG. 16

BNSDOCID: <WO_____0131900A2_I_>

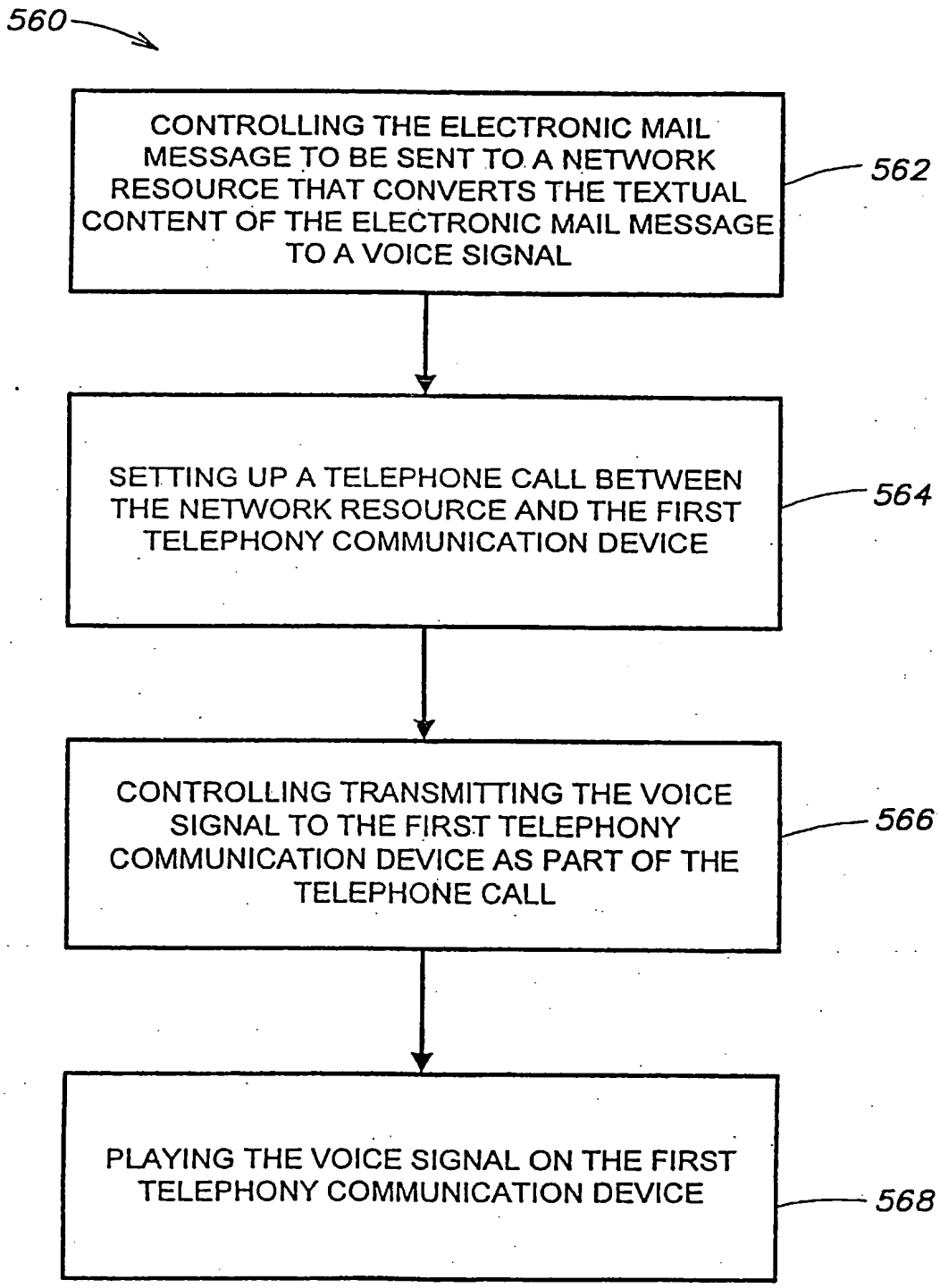


FIG. 17

BNSDOCID: <WO_____0131900A2_I_>

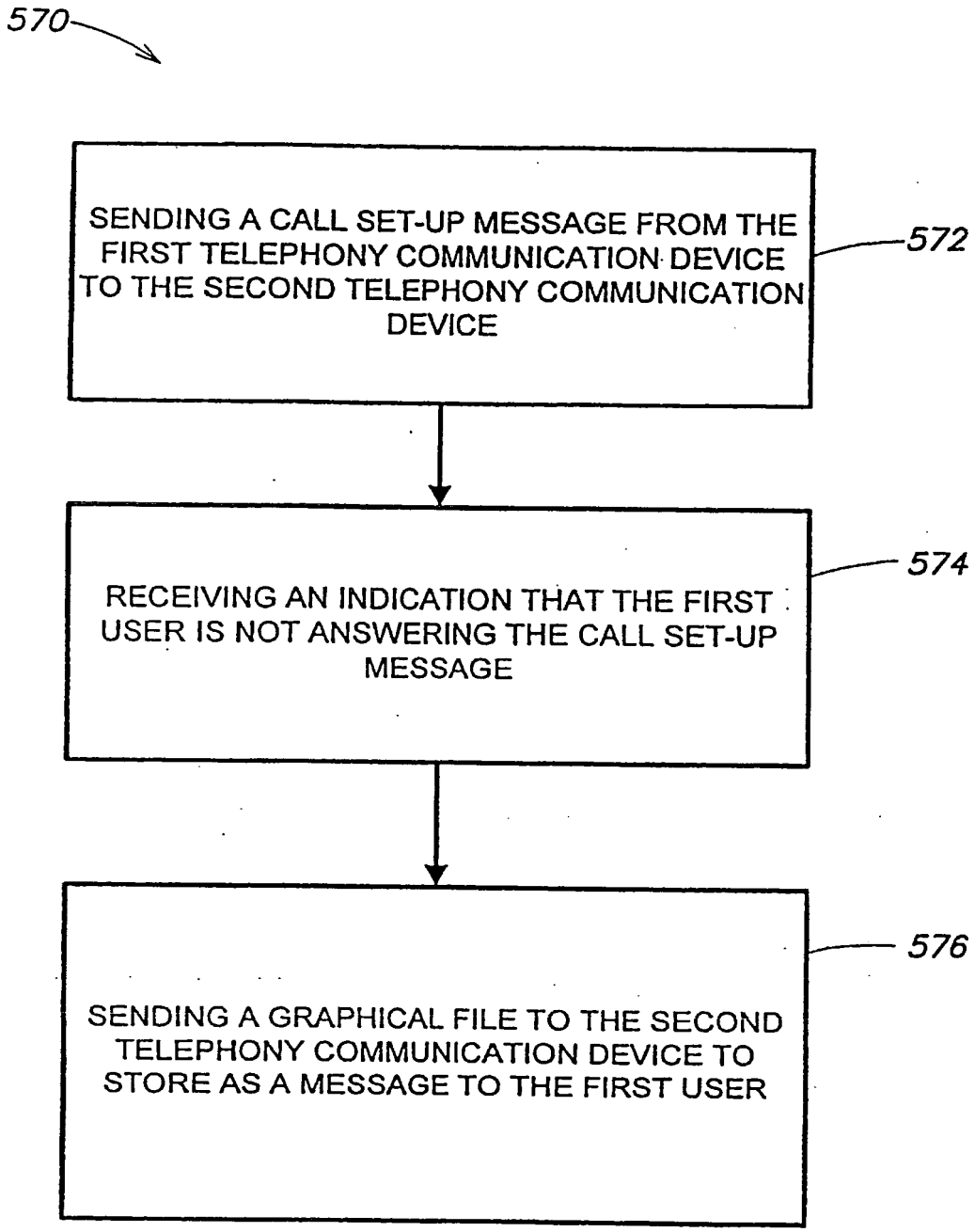


FIG. 18

BNSDOCID: <WO_____0131900A2_I_>

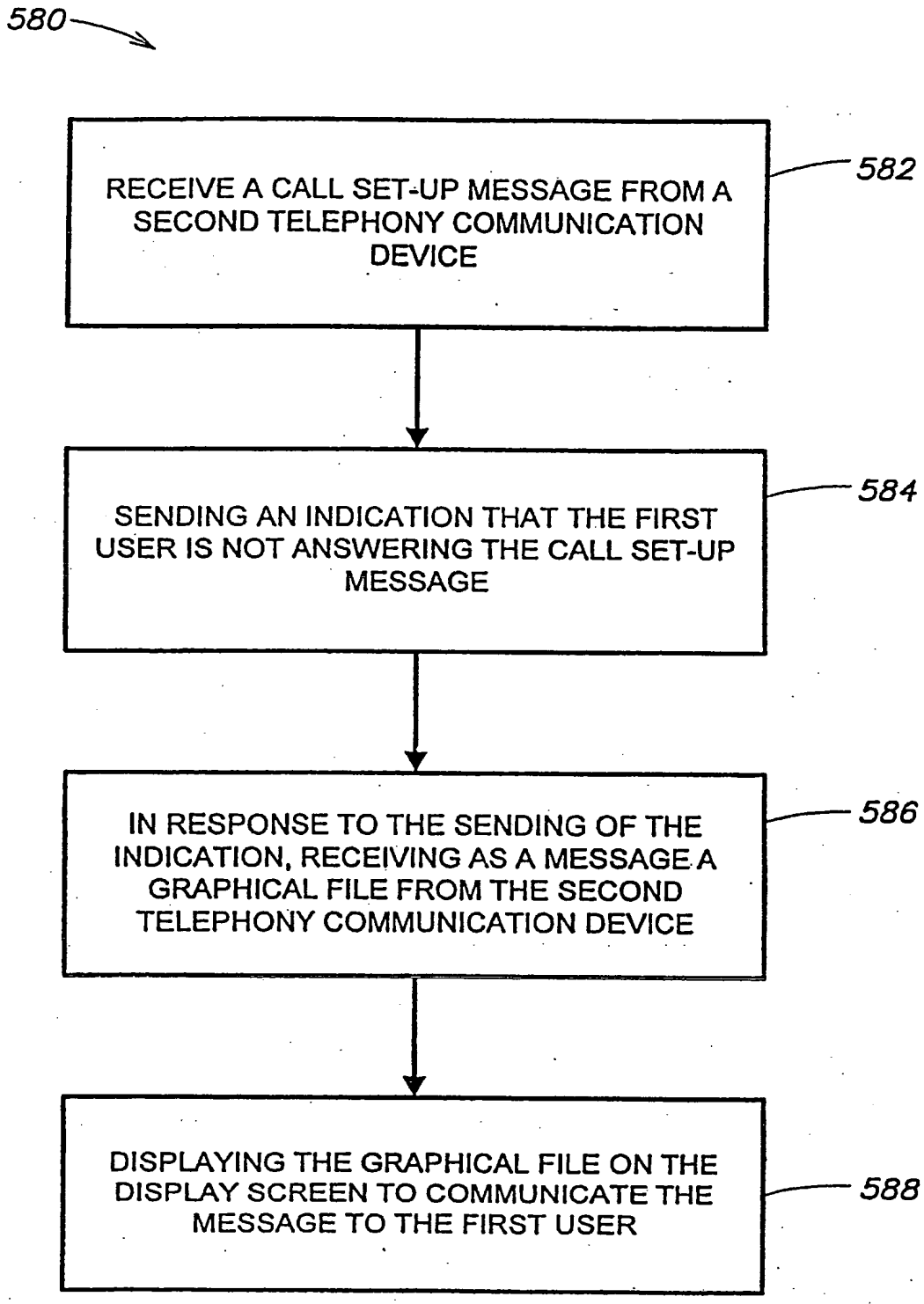


FIG. 19

BNSDOCID: <WO_____0131900A2_I_>

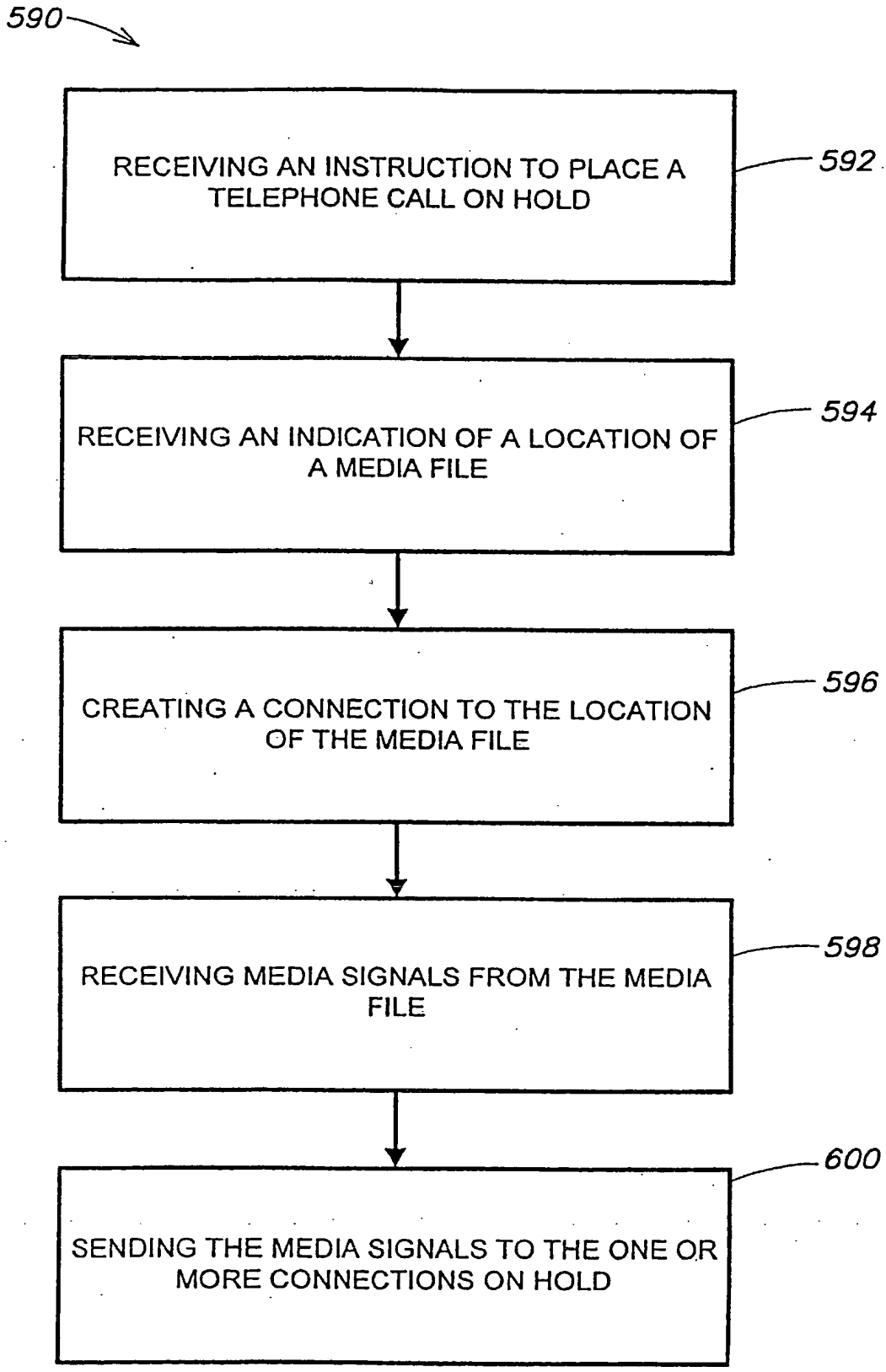


FIG. 20

BNSDOCID: <WO_____0131900A2_I_>

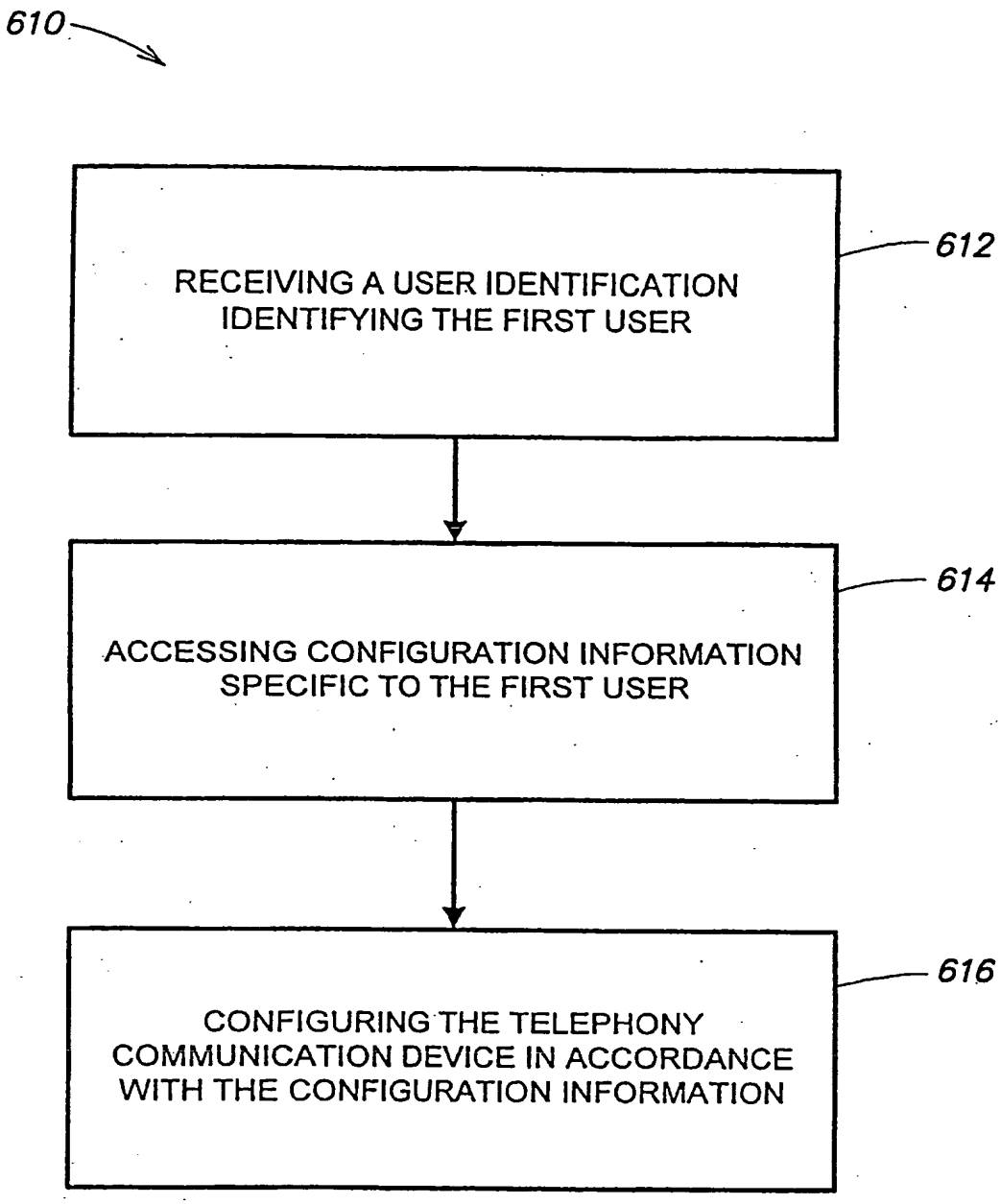


FIG. 21

BNSDOCID: <WO_____0131900A2_I_>

THIS PAGE BLANK (USPTO)

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☑ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)